

# **Pengembangan Sistem Pendeteksi Kehadiran Pegawai Berbasis *Fingerprint* Wifi**

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Rizky Haryandi Rahman

NIM: 145150201111074



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PENGEMBANGAN SISTEM PENDETEKSI KEHADIRAN PEGAWAI BERBASIS  
FINGERPRINT WIFI

### SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Rizky Haryandi Rahman  
NIM: 145150201111074

Skripsi ini telah diuji dan dinyatakan lulus pada  
23 Maret 2018

Telah diperiksa dan disetujui oleh:

Pembimbing I

Adhitya Bhawiyuga, S.Kom, M.S  
NIK: 201405 890720 1 001

Pembimbing II

Ir. Primantara Hari Trisnawan, M. Sc  
NIP: 19880912 199403 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Dr. Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001

### PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 04 April 2018



Rizky Haryandi Rahman

NIM: 145150201111074



## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT, karena berkat rahmat serta bimbingan-Nya, penulis dapat menyelesaikan penulisan skripsi dengan baik. Penulisan skripsi ini diajukan untuk memenuhi sebagian persyaratan untuk memperoleh gelar Sarjana Komputer pada Proram Studi Teknik Informatika Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya. Judul yang penulis ajukan adalah Implementasi Sistem Pendeteksi Kehadiran Pegawai Berbasis *Fingerprint* Wifi.

Dalam penyusunan dan penulisan skripsi ini tidak terlepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis dengan senang hati menyampaikan terima kasih kepada:

1. Bapak H. Rahmansyah dan Ibu Hj. Rusnani selaku orang tua penulis yang selalu memberikan dukungan dan doa untuk penulis agar dapat menyelesaikan skripsi ini,
2. Bapak Adhitya Bhawiyuga, S.Kom, M.Sc dan Ir. Primantara Hari Trisnawan, M. Sc selaku dosen pembimbing I dan pembimbing II yang telah dengan sabar membimbing, mengarahkan, serta meluangkan waktu untuk penulis sehingga dapat menyelesaikan skripsi ini,
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D, Bapak Agus Wahyu Widodo, S.T, M.Sc dan Bapak M. Tanzil Furqon, S.Kom, M.CompSc selaku Ketua Jurusan Teknik Informatika, Ketua Program Studi Teknik Informatika dan Sekertaris Jurusan Teknik Informatika,
4. Bapak Drs. Marji, MT selaku Dosen Pembimbing Akademik yang dengan sabar mengarahkan serta memantau kegiatan akademik penulis pada tiap semester,
5. Seluruh Dosen Program Studi Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis,
6. Seluruh Civitas Akademik Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini,
7. Teman–teman angkatan 2014 Informatika, teman – teman F.Zoo, Harits Abdulrohman, Chandra Yogi Adhitama, serta para senior, terima kasih atas segala bantuan selama pengerjaan skripsi dan selama menempuh studi di Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya,
8. Syifa Adeliya Oktaviani yang selalu memberikan dukungan serta motivasi untuk penulis,
9. Seluruh pihak yang telah membantu kelancaran penulisan skripsi yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan baik format penulisan maupun isinya. Oleh karena itu, saran dan kritik membangun dari para pembaca senantiasa penulis harapkan guna perbaikan bagi skripsi selanjutnya. Semoga skripsi ini dapat memberikan manfaat bagi semua pihak.

Malang, 04 April 2018

Penulis

[rizkyharyandir@yahoo.co.id](mailto:rizkyharyandir@yahoo.co.id)



## ABSTRAK

Melakukan pemeriksaan kehadiran pegawai merupakan hal penting bagi setiap institusi atau perusahaan untuk memeriksa kinerja pegawai yang dimilikinya. Setiap institusi atau perusahaan pasti memiliki cara tersendiri dalam melakukan pemeriksaan kehadiran pegawai, seperti cara manual dengan menggunakan kertas, sistem aplikasi di *smartphone*, RFID (*Radio Frequency Identification*), dan ada yang sudah menggunakan teknik biomentrik. Beberapa cara tersebut memang dapat memudahkan perusahaan dalam melakukan pendataan kehadiran pegawai, tetapi sistem tersebut masih memiliki beberapa kekurangan, contohnya pada sistem presensi menggunakan sidik jari, sidik jari terkadang tidak dapat terbaca dengan baik oleh sistem, pegawai harus mencapai lokasi tertentu untuk melakukan presensi, hal ini yang terkadang membuat pegawai sering menunda atau terlambat melakukan presensi, akibatnya pegawai lupa dan tidak melakukan presensi. Berdasarkan permasalahan yang terjadi, maka pada penelitian kali ini akan di bangun sebuah sistem pendeteksi kehadiran pegawai berbasis *fingerprint* wifi, *fingerprint* wifi yang dimaksud adalah sebuah alamat MAC yang tertangkap oleh router mikrotik ketika *smartphone* terkoneksi. Sistem akan membaca alamat MAC dan *interface* dari router mikrotik melalui API untuk RouterOS, kemudian menyimpan kedalam *database*, dan membandingkannya dengan alamat MAC yang sudah ada pada *database*, sehingga pada saat *smartphone* pegawai terkoneksi ke sebuah router atau *access point* yang telah ditentukan, maka pegawai akan di anggap hadir dan akan diketahui keberadaannya berdasarkan tempat *smartphone* terkoneksi. Hasil pengujian fungsional dan performa menunjukan keseluruhan fungsi pada sistem ini berjalan dengan baik dan waktu yang dibutuhkan sistem untuk membaca dari router memiliki rata-rata keseluruhan kurang dari 1 detik, dan waktu yang diperlukan sistem untuk mendeteksi kehadiran beserta lokasi pegawai kurang lebih 5 detik.

Kata kunci: sistem presensi, alamat MAC, router *mikrotik*, *fingerprint*.



## ABSTRACT

*Checking the employees attendance is an important matter for every institution or company to see their own employee performance. Each instituion or company must have their own way to check the employees attendance, such as using a paper, smartphone application, RFID (Radio Frequency Identification), and biomentric technique. Those kind of ways are really making the company easier to check the employees attendance, but there is a flaw, for example is on the fingerprinting system when using the fingerprint, sometimes the fingerprint can not be detected by the system, moreover an employee should reach a specific location to make a presence. This sometimes make the employees to procrastinate or late to make a presence. As a result, the employees often forget and do not make a presence. From the problem, this reasearch is made to make a presence system for employees using a wifi fingerprint, wifi fingerprint means a MAC address which is caught by microtic router when a smartphone is connected. System will read the MAC address and interface of the router mikrotik through the API for RouterOS, then save it into the database, and compare it with the MAC address already in the database, so when it happens, an employee will be considered as present and the location of an employee will be known based on a place where a smartphone is connected. The testing result of functionality and performance shows the all functions on system run well and the time that needed by a system to read from a router has an average below of one second and the time for system to detect the presence including the employees location is about five seconds.*

*Keywords: presence system, MAC address, router mikrotik, fingerprint.*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN .....	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 KAJIAN PUSTAKA.....	5
2.2 DASAR TEORI.....	6
2.2.1 Wifi .....	6
2.2.2 <i>Fingerprint Wifi</i> .....	7
2.2.3 API Untuk RouterOS .....	7
2.2.4 <i>Database</i> .....	8
2.2.5 Sistem Presensi .....	9
BAB 3 METODOLOGI .....	10
3.1 Studi Literatur .....	10
3.2 Analisis Kebutuhan .....	11
3.3 Perancangan Sistem.....	12
3.3.1 Desain Alur Kerja Sistem .....	12
3.4 Implementasi Sistem .....	13



3.5 Pengujian dan Analisis .....	13
3.6 Kesimpulan.....	14
BAB 4 Analisis Kebutuhan .....	15
4.1 Deskripsi Umum.....	15
4.1.1 Perspektif Produk/Sistem .....	15
4.1.2 Kegunaan.....	15
4.1.3 Karakteristik Pengguna .....	15
4.1.4 Lingkungan Operasi.....	15
4.1.5 Batasan Perancangan dan Implementasi.....	15
4.1.6 Asumsi dan Ketergantungan .....	16
4.2 Kebutuhan Antarmuka Eksternal.....	16
4.2.1 Antarmuka Pengguna.....	16
4.2.2 Antarmuka Perangkat Keras .....	16
4.2.3 Antarmuka Perangkat Lunak.....	16
4.2.4 Antarmuka Komunikasi.....	17
4.3 Kebutuhan Fungsional .....	17
BAB 5 Perancangan Dan Implementasi.....	19
5.1 Perancangan Sistem.....	19
5.1.1 Gambaran Umum Sistem .....	19
5.1.2 Alur Kerja Sistem .....	20
5.1.3 Perancangan Perangkat Lunak.....	21
5.1.4 Perancangan Topologi Jaringan .....	28
5.1.5 Perancangan <i>Database</i> .....	28
5.2 Implementasi Sistem .....	29
5.2.1 Mendeteksi Kehadiran Pegawai.....	29
5.2.2 Mendeteksi Lokasi Pegawai .....	33
5.2.3 Menambah Data Pegawai .....	34
5.2.4 Menampilkan Daftar Pegawai.....	36
5.2.5 Menghapus Data Pegawai.....	38
5.2.6 Edit Data Pegawai .....	38
5.2.7 Tampil <i>History</i> Pegawai.....	41
BAB 6 PENGUJIAN DAN ANALISIS.....	47

6.1 Pengujian Fungsional .....	47
6.2 Pengujian Performa .....	50
BAB 7 Penutup .....	58
7.1 Kesimpulan.....	58
7.2 Saran .....	59
DAFTAR PUSTAKA.....	60
LAMPIRAN .....	61



## DAFTAR TABEL

Tabel 2.1 Sourcecode <i>API Router</i> .....	7
Tabel 4.1 Kebutuhan fungsional .....	18
Tabel 5.1 Data karyawan .....	28
Tabel 5.2 Data pegawai .....	29
Tabel 5.3 <i>Contoller</i> koneksi ke router .....	29
Tabel 5.4 Controller insert alamat MAC .....	30
Tabel 5.5 Model menampilkan kehadiran .....	31
Tabel 5.6 <i>Controller</i> menampilkan kehadiran .....	31
Tabel 5.7 <i>View</i> halaman kehadiran .....	31
Tabel 5.8 <i>Contoller</i> mendeteksi lokasi .....	33
Tabel 5.9 Deteksi lokasi pegawai .....	33
Tabel 5.10 <i>View</i> form input data pegawai .....	34
Tabel 5.11 Model menambahkan pegawai .....	35
Tabel 5.12 Controller menambahkan data pegawai .....	35
Tabel 5.13 Model menampilkan data pegawai .....	36
Tabel 5.14 <i>Contoller</i> menampilkan data pegawai .....	36
Tabel 5.15 <i>View</i> menampilkan data pegawai .....	36
Tabel 5.16 <i>View</i> delete data pegawai .....	38
Tabel 5.17 Model delete data pegawai .....	38
Tabel 5.18 Controller delete data pegawai .....	38
Tabel 5.19 Model menampilkan data pegawai yang akan diedit .....	38
Tabel 5.20 <i>View</i> menampilkan form edit .....	39
Tabel 5.21 Model edit data pegawai .....	40
Tabel 5.22 <i>Contoller</i> edit data pegawai .....	41
Tabel 5.23 Model menampilkan <i>history</i> kehadiran .....	41
Tabel 5.24 <i>View</i> menampilkan <i>history</i> kehadiran .....	42
Tabel 5.25 <i>Contoller</i> menampilkan <i>history</i> pegawai .....	43
Tabel 5.26 <i>View history</i> pegawai sesuai nama .....	43
Tabel 5.27 Controller <i>history</i> pegawai sesuai nama .....	45
Tabel 5.28 Model <i>history</i> pegawai sesuai nama .....	45

Tabel 6.1 Skenario pengujian fungsional .....	47
Tabel 6.2 Hasil pengujian fungsional .....	49
Tabel 6.3 Skenario pengujian performa.....	50
Tabel 6.4 Hasil pengujian PP_001 .....	52
Tabel 6.5 Hasil pengujian PP_002 .....	54
Tabel 6.6 Hasil pengujian PP_003 .....	56
Tabel 6.7 Hasil pengujian PP_004 .....	57



## DAFTAR GAMBAR

Gambar 3.1 Diagram alir metodologi penelitian .....	10
Gambar 3.2 Desain alur kerja sistem .....	12
Gambar 4.1 Antar muka komunikasi .....	17
Gambar 5.1 Konsep umum Sistem Pendeteksi Kehadiran .....	19
Gambar 5.2 Alur Kerja Sistem .....	20
Gambar 5.3 Mendeteksi kehadiran .....	21
Gambar 5.4 Mendeteksi lokasi .....	22
Gambar 5.5 Alur kerja menambahkan pegawai .....	23
Gambar 5.6 Alur kerja menampilkan data pegawai .....	24
Gambar 5.7 Alur kerja edit data pegawai .....	25
Gambar 5.8 Alur kerja hapus data pegawai .....	26
Gambar 5.9 Alur kerja menampilkan <i>history</i> berdasarkan tanggal dan nama .....	27
Gambar 5.10 Perancangan topologi jaringan .....	28
Gambar 6.1 Hasil capture wireshark pengujian PP_001.....	52
Gambar 6.2 Follow TCP stream pengujian PP_001.....	52
Gambar 6.3 Grafik hasil pengujian.....	53
Gambar 6.4 Hasil capture wireshark pengujian PP_002.....	54
Gambar 6.5 Hasil capture wireshark pengujian PP_002.....	54
Gambar 6.6 Grafik hasil pengujian PP_002.....	55
Gambar 6.7 Hasil capture wireshark pengujian PP_002.....	55
Gambar 6.8 Grafik hasil pengujian PP_003.....	56
Gambar 6.9 Grafik hasil pengujian PP_004.....	57

## DAFTAR LAMPIRAN

Lampiran 1 Bukti Hasil Pengujian Fungsional .....	61
Lampiran 2 <i>Library</i> API RouterOS.....	65
Lampiran 2 <i>Library</i> API RouterOS.....	76





## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Melakukan pemeriksaan kehadiran pegawai merupakan hal penting bagi setiap institusi atau perusahaan untuk memeriksa kinerja pegawai yang dimilikinya. Tidak hadirnya pegawai tentu berdampak pada tingkat kinerja suatu perusahaan. Walaupun untuk menyelesaikan pekerjaan dapat dilakukan di rumah, tapi banyak perusahaan yang memilih pegawainya tetap hadir agar dapat memantau pekerjaan yang dilakukan pegawai tersebut. Untuk menyelesaikan masalah tersebut banyak perusahaan yang mencoba menggunakan sebuah sistem presensi untuk pegawainya.

Banyak cara yang dapat dilakukan untuk memeriksa kehadiran pegawai seperti menggunakan cara manual dengan menggunakan kertas, sistem aplikasi di *smartphone*, RFID (*Radio Frequency Identification*), dan beberapa telah memanfaatkan teknik biometrik sebagai sistem presensi. Cara-cara seperti di atas memang dapat memudahkan dari sisi perusahaan dalam melakukan pendataan kehadiran pegawainya, tetapi masih memiliki beberapa kekurangan, contohnya pada sistem presensi menggunakan sidik jari, sidik jari terkadang tidak dapat terbaca dengan baik oleh sistem, belum lagi pegawai harus mencapai lokasi tertentu untuk melakukan presensi, ini yang terkadang membuat pegawai sering menunda atau terlambat melakukan presensi, akibatnya pegawai sering lupa dan tidak melakukan presensi.

Pada penelitian sebelumnya yang berjudul "*Student Attendance Recording System Using Face Recognition with GSM Based*" yang dilakukn Patil, dkk (2014), membahas tentang sistem kehadiran siswa berdasarkan teknologi pengenalan wajah dan *otentikasi*. Penelitian yang dilakukan Patil, dkk (2014) mengenali wajah siswa melalui webcam, kemudian membandingkannya dengan gambar yang ada pada *database* dengan melakukan pengukuran kesamaan menggunakan jarak antara titik kontrol dari gambar yang masuk. Jika gambar sesuai maka dianggap telah hadir. Pada penelitian tersebut berhasil mendeteksi kehadiran dengan mendeteksi wajah melalui webcam, akan tetapi sistem ini bergantung pada seberapa bagus kamera dalam menangkap wajah siswa, hal tersebut memungkinkan bahwa, jika kamera tidak mampu menangkap gambar dengan jelas, maka sistem tidak dapat mendeteksi kehadiran siswa tersebut.

Pada penelitian kali ini dibangun sebuah sistem pendeteksi kehadiran pegawai berbasis *fingerprint wifi*. Sistem memanfaatkan *fingerprint wifi* untuk mendeteksi kehadiran pegawai, *fingerprint wifi* yang dimaksud adalah sebuah alamat MAC pada *smartphone*, alamat MAC bersifat unik sehingga dapat digunakan untuk mengidentifikasi sebuah *smartphone* yang terkoneksi pada router atau *access point*. Sistem akan mengenali alamat MAC *smartphone* pegawai yang terkoneksi ke router atau *access point*, kemudian menyimpannya ke dalam *database* lalu membandingkannya dengan data yang ada pada *database*, sistem juga dapat mendeteksi lokasi pegawai pada sebuah gedung melalui router

atau *access point* yang terkoneksi pada *smartphone*. Sistem ini menyediakan *interface* berbasis web karena mudah di akses tanpa harus menggunakan aplikasi tambahan dan mudah jika digabungkan dengan sistem lain seperti web institusi atau perusahaan. Pada penelitian kali ini akan dilakukan pengujian fungsional dan waktu yang dibutuhkan sistem untuk mendeteksi kehadiran beserta lokasi pegawai, agar dapat diketahui apakah semua fungsi berjalan dengan baik dan berapa lama waktu yang dibutuhkan sistem untuk mendeteksi kehadiran dan lokasi pegawai.

## 1.2 Rumusan masalah

Berdasarkan uraian latar belakang di atas, maka dapat dirumuskan permasalahan pada penelitian ini yaitu sebagai berikut :

1. Bagaimana sistem membaca data alamat MAC dan lokasi pegawai dari router?
2. Bagaimana sistem dapat mendeteksi kehadiran pegawai dan lokasi pegawai berdasarkan data yang diambil dari router?
3. Bagaimana kinerja fungsional dan waktu yang dibutuhkan sistem untuk mendeteksi kehadiran beserta lokasi pegawai?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah terbentuknya sebuah sistem pendeteksi kehadiran pegawai berbasis *fingerprint* wifi, dimana sistem dapat mendeteksi kehadiran seorang pegawai hanya dengan sebuah perangkat yang dibawanya, sistem juga mampu mendeteksi posisi pegawai melalui router atau *access point* yang terkoneksi pada perangkatnya. Pada sistem ini data kehadiran pegawai tersimpan dalam database, sehingga dapat membantu menangani pendataan kehadiran pegawai pada perusahaan.

## 1.4 Manfaat

Penelitian ini diharapkan mampu membantu perusahaan untuk mengatur sistem presensi pegawai, dan memudahkan pegawai dalam melakukan presensi, sehingga tidak ada pegawai yang lupa melakukan presensi kehadiran. Sistem ini diharapkan dapat membantu mengolah data presensi pegawai dengan lebih mudah.

## 1.5 Batasan masalah

Agar pembahasan dalam penelitian ini dapat dilakukan secara fokus dan dapat mendapatkan hasil yang sesuai dengan yang diharapkan, maka perlu diterapkan batasan permasalahan antara lain :

1. Sistem menggunakan *database* mysql pada phpmyadmin.
2. Sistem menggunakan router mikrotik, sebagai router utama.

3. Sistem ini difokuskan untuk mendeteksi kehadiran pegawai lewat *smartphone*.
4. Sistem hanya dapat melakukan proses presensi menggunakan alamat MAC pada *smartphone* pegawai.
5. Sistem hanya mampu melakukan presensi dan mendeteksi keberadaan pegawai jika *smartphone* pegawai terkoneksi pada router atau *access point* yang telah terhubung dengan sistem.
6. Alamat MAC pada *smartphone* harus bersifat unik, dalam artian bahwa alamat MAC tidak boleh diubah.
7. Sistem tidak dapat mengatasi kecurangan presensi, seperti *smartphone* sengaja ditinggalkan dan lain sebagainya.

## 1.6 Sistematika pembahasan

Sistematika pembahasan penelitian ini disusun sebagai berikut:

### BAB 1 PENDAHULUAN

Pada bagian pendahuluan, menjelaskan tentang latar belakang dilakukannya penelitian ini, terutama menjelaskan pentingnya sistem presensi bagi institusi dan perusahaan, serta menjelaskan secara singkat penelitian terdahulu, kemudian menjelaskan rumusan masalah, tujuan, serta manfaat dari penelitian yang dilakukan.

### BAB 2 LANDASAN KEPUSTAKAAN

Bab ini akan diuraikan mengenai dasar teori yang digunakan dalam proses penelitian, beserta penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang dan metode yang berkaitan pada penelitian ini. .

### BAB 3 METODOLOGI

Bab ini menjelaskan mengenai langkah-langkah dalam melaksanakan penelitian, menguraikan mengenai metode serta langkah kerja penelitian yang terdiri atas studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian serta pengambilan kesimpulan dari penelitian yang telah dilakukan.

### BAB 4 ANALISI KEBUTUHAN

Bab ini menjelaskan mengenai proses analisis kebutuhan sistem yang diperlukan meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan fungsional

### BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini akan menjelaskan perancangan sistem dan mengimplementasikan hasil dari perancangan sistem.

## **BAB 6 PENGUJIAN DAN ANALISIS**

Bab ini akan memaparkan hasil analisis dari pengujian yang dilakukan pada penelitian ini.

## **BAB 7 PENUTUP**

Bab ini menjelaskan kesimpulan yang diperoleh dari perancangan, implementasi, dan pengujian sistem, serta saran untuk pengembangan penelitian selanjutnya.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi landasan kepustakaan yang berisi kajian pustaka dan dasar teori dalam menunjang proses penelitian. Kajian pustakan ini membahas terkait penelitian-penelitian yang sudah ada dan mendukung dalam proses penelitian yang diusulkan. Dasar teori ini berisi berbagai teori yang diperlukan dalam penelitian ini.

### 2.1 KAJIAN PUSTAKA

Kajian pustaka membahas berbagai penelitian yang telah dilakukan sebelumnya, sehingga menjadi pendukung dalam proses penelitian yang diusulkan. Pada penelitian ini, kajian pustaka diambil dari beberapa penelitian yang sudah ada.

Penelitian yang dilakukan oleh Narra, dkk (2017) yang berjudul *"Aadhaar Based Biometric Attendance System Using Wireless Fingerprint Terminals"*, menghadirkan sistem kehadiran dengan menggunakan *wireless fingerprint terminal* (WTF), yang berisi pemindai sidik jari, GSM (*Global System for Mobile Communications*) untuk konektivitas *web* melalui transmisi data dari perangkat ke server dan sebaliknya. Sistem ini menggunakan otentifikasi biometrik secara *online*. Setiap kali siswa melakukan presensi menggunakan sidik jari melalui WTF, maka data berupa sidik jari akan dikirimkan ke server oleh WTF, sidik jari akan dibandingkan dengan sidik jari yang ada di *database* WTF, jika verifikasi berhasil maka kehadiran siswa akan dicatat beserta tanggal dan waktu kehadiran (Narra,dkk., 2017).

Penelitian yang dilakukan oleh Patil, dkk (2014) yang berjudul *"Student Attendance Recording System Using Face Recognition with GSM Based"*, membahas tentang kehadiran siswa berdasarkan teknologi pengenalan wajah dan otentikasi. Pada penelitian ini dijelaskan bahwa sistem ini akan mengenali wajah siswa melalui webcam, kemudian membandingkannya dengan gambar yang ada pada *database* dengan melakukan pengukuran kesamaan menggunakan jarak antara titik kontrol dari gambar yang masuk. Jika gambar sesuai maka akan dianggap telah hadir.

Penelitian yang dilakukan oleh Grag, dkk (2012) yang berjudul *"MAC and Logical addressing"* menyatakan bahwa alamat MAC merupakan sebuah alamat fisik yang memiliki panjang 48 bit yang terletak pada lapisan *datalink* dari model OSI, dari 48 bit tersebut dibagi menjadi kode vendor 24 bit, dan alamat serial 24 bit, kode ini unik untuk setiap sistem. Alamat MAC biasanya mengkodekan nomor identifikasi produsen. Alamat MAC juga dikenal sebagai Ethernet Hardware Address (EHA), alamat perangkat keras, alamat adaptor, atau alamat fisik. Pada penelitian ini menganalisis bahwa alamat MAC unik untuk setiap perangkat.

Pada penelitian yang dilakukan oleh Gencho, dkk (2014) yang berjudul *"One implementation of API interface for RouterOS"* menyajikan implementasi antarmuka API untuk router mikrotik, yang memungkinkan penggunaan skrip PHP



untuk mendapatkan data untuk konfigurasi dan pengelolaan perutean perangkat yang menggunakan OS ini. Penelitian ini menunjukkan bahwa RouterOS dapat terintegrasi dengan sistem eksternal melalui antarmuka API (*Application Programmable Interface*). Hal ini menguntungkan karena memungkinkan pengguna berkomunikasi dengan router untuk mendapatkan informasi, konfigurasi dan pengelolaan perangkat tersebut. Layanan yang menyediakan antarmuka ini disebut api dan, protokol yang digunakan disebut sebagai RouterOS API. Penelitian ini menunjukan bahwa API mampu untuk berinteraksi langsung dengan RouterOS. Beberapa contoh API dari peneliti seperti, menunjukkan log dari router, melakukan ping dari router, menyimpan kemudian menganalisis lalu lintas jaringan, dan menampilkan alamat MAC *client*, API yang dibuat peneliti harus dilengkapi dengan perpustakaan pendukung yaitu Autoload.php.

## 2.2 DASAR TEORI

Penulisan dasar teori membahas berbagai teori dasar untuk menunjang proses penelitian. Pada sub bab dasar teori akan dijelaskan referensi dan teori dasar pendukung mengenai gambaran umum tentang sistem yang akan dibuat.

### 2.2.1 Wifi

Wifi adalah sebuah media transmisi di dalam jaringan komputer, khususnya *wireless*, yang berdasarkan pada standarisasi internasional IEEE 802.11 (Putu, 2015). Wifi memanfaatkan peralatan elektronik untuk bertukar data secara nirkabel melalui sebuah jaringan komputer, termasuk koneksi internet. Pada sejarahnya wifi memiliki beberapa standarisasi keamanan, *standarisasi* keamanan tersebut meliputi:

a. WEP (*Wired Equivalent Privacy*)

Merupakan *standarisasi* keamanan pertama yang digunakan wifi. WEP masih menggunakan kunci yang bersifat statis, sehingga mudah dibobol. Algoritma RC4 yang digunakan memiliki banyak kelemahan terkait keamanan, dan CRC (*Cyclic Redundance Check*) dan IV (*Initializaion Vector*) memiliki banyak masalah di dalamnya.

b. WPA (*WiFi Protected Access*)

Merupa *standarisasi* keamanan yang merupakan pengembangan dari WEP, dimana dibuat lebih sederhana dan sulit untuk ditembus.

c. WPA PSK (*Pre Shared Key*)

Merupakan *standarisasi* kemanan, di mana di dalamnya tidak diperlukan adanya server untuk otentifikasi pengguna.

d. WPA2

Merupakan versi kedua dari WPA, yang dirancang untuk memperbaiki kelemahan dari WPA generasi sebelumnya. WPA2 personal adalah salah satu dari dua variasi dari protokol WPA2, di mana sesuai untuk digunakan dalam pengaturan kelas bisnis atau rumahan.



### 2.2.2 Fingerprint Wifi

*Fingerprint* wifi yang dimaksud disini adalah sebuah alamat MAC yang tertangkap oleh router atau access point ketika sebuah *smartphone* terkoneksi, alamat MAC bersifat unik sehingga dapat digunakan untuk mengidentifikasi sebuah *smartphone* yang terkoneksi pada router. Alamat MAC adalah sebuah alamat jaringan yang diimplementasikan pada lapisan datalink pada model OSI (*Open System Interconnection*) yang digunakan untuk merepresentasikan sebuah node tertentu dalam jaringan. Alamat MAC berhubungan dengan pengalamatan secara fisik komputer di dalam jaringan (Putu, 2015). Dalam sebuah jaringan berbasis Ethernet, Alamat MAC merupakan alamat yang unik yang memiliki panjang 48-bit (6 byte) yang mengidentifikasi sebuah komputer, *interface* dalam sebuah router, atau node lainnya dalam jaringan (Al-Husainy, 2014).

### 2.2.3 API Untuk RouterOS

API (*Application Programing Interface*) merupakan kumpulan dari beragam perintah, fungsi, dan protocol pada sebuah jaringan, yang bekerjadalam menjadikan aplikasi agar dapat berkomunikasi dengan sistem operasi, perangkat keras dalam jaringan komputer (Putu, 2015). API berada pada dua sisi *entitas*, yaitu berjalan pada sisi *Applicaition Layer* dan berjalan pada sisi sistem operasi, hal ini memudahkan untuk mengembangkan sebuah aplikasi agar dapat berkomunikasi dengan baik dengan sistem operasi. API (*Application Programing Interface*) memudahkan komunikasi yang meliputi *socket interface*, TLI (*Transport Layer Interface*), dan *Stream*.

RouterOS adalah sistem operasi yang digunakan oleh perusahaan MikroTik dalam implementasi perangkat routing khusus, dan untuk mengubah perangkat computer standar menjadi router (Stoitsov, G., Rongelov,V. 2014). Sistem ini menyediakan jaringan yang kaya fitur, salah satu fiturnya adalah dapat terintegrasi dengan sistem eksternal melalui API (*Application Programmable Interface*) *interface*. Hal ini memungkinkan implementasi perangkat lunak pengguna untuk berkomunikasi dengan router untuk mendapatkan informasi, konfigurasi dan pengelolaan perangkat tersebut. Layanan yang menyediakan kesempatan ini disebut API dan protokol yang digunakan disebut sebagai RouterOS. Mikrotik menyediakan API dengan berbagai bahasa pemrograman salah satunya adalah PHP, seperti yang diimplementasikan pada penelitian ini.

Berikut ini adalah salah satu contoh API RouterOS yang dibuat untuk menampilkan alamat MAC yang tentunya menjadi dasar dari penelitian ini, untuk mengimplementasikan API ini mikrotik menyediakan *library* khusus agar API ini dapat terhubung dengan router.

**Tabel 2.1 Sourcecode API Router**

1	<?php
2	
3	

```

4 use PEAR2\Net\RouterOS; //inclusion of the library
5 require_once 'PEAR2/Autoload.php';//
6 include Autoload.php
7 header ('Content-Type: text/html;charset=UTF-8');?>
8 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
9 Strict//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1strict.dt
10 d">
11 <html xmlns="http://www.w3.org/1999/xhtml">
12 <head>
13 <title>YourAlamat MAC</title>
14 </head>
15 <body>
16 <h1>
17 <?php try {
18 $client = new RouterOS\Client ('192.168.0.1', 'admin',
19 'password');
20 $printRequest = new RouterOS\Request ('/iparp print
21 .proplist=mac-address'); $printRequest>setQuery
22 (RouterOS\Query::where ('address',
23 $_SERVER['REMOTE_ADDR']));
24 $mac=$client>sendSync ($printRequest)>
25 getArgument ('mac-address');
26 if (null!=$mac){
27 echo'YourAlamat MAC: ', $mac;}
28 else{echo'YourIPaddress ('', $_SERVER['REMOTE_ADDR'],') is
29 not part of our network, so we can not determine Alamat MAC';
30 }
31 }
32 catch (Exception $e)
33 {
34 }?> </h1> </body> </html>

```

Sumber : (Stoitsov, G., Rongelov,V., 2014)

## 2.2.4 Database

*Database* atau yang biasa di sebut basis data terdiri dari 2 kata, yaitu data dan base. Base (basis) memiliki arti sebagai gudang, markas, atau tempat berkumpul, sedangkan data, merupakan representasi fakta duna nyata mewakili suatu objek seperti manusia, barang, hewan dan lain-lain, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya (Fathansyah, 2012). *Database* merupakan kumpulan informasi yang tersimpan di dalam komputer secara sistematis sehingga bisa diperiksa dengan menggunakan program komputer untuk mendapatkan informasi dari *database*. Perangkat lunak yang digunakan untuk mengelola dan memanggil query (query) *database* disebut sistem manajemen *database* (Imam, 2016 ). *Database* merupakan aspek penting dalam sistem informasi, *database* adalah tempat penyimpanan data diproses lebih lanjut *database* penting karena bisa mengatur data, menghindari duplikasi data, hubungan antara data yang tidak jelas dan juga pembaruan yang rumit.

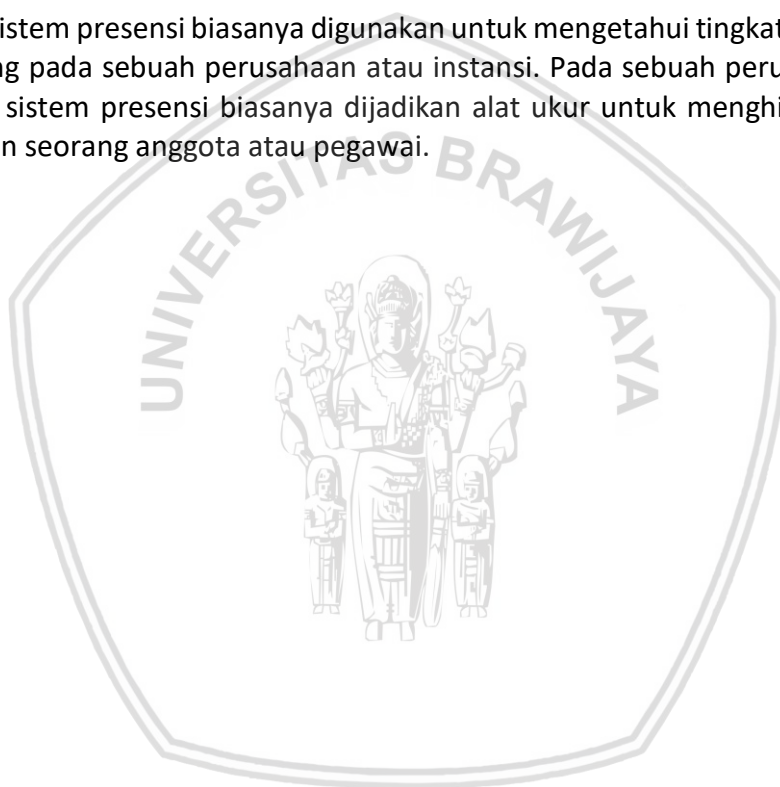
Mysql merupakan salah satu perangkat lunak yang digunakan untuk manajemen basis data, merupakan turunan dari SQL (*Structured Query Language*) salah satu konsep utama dalam *database*, SQL adalah sebuah pengoperasian

*database* untuk pemilihan atau seleksi pemasukan data, yang memungkinkan pengoperasian data dikerjakan secara otomatis. Salah satu keistimewaan mysql adalah dapat diadministrasikan ke dalam bentuk web menggunakan phpmyadmin. Perintah-perintah SQL juga dapat dikombinasikan dengan bahasa pemrograman lain yaitu PHP .

### 2.2.5 Sistem Presensi

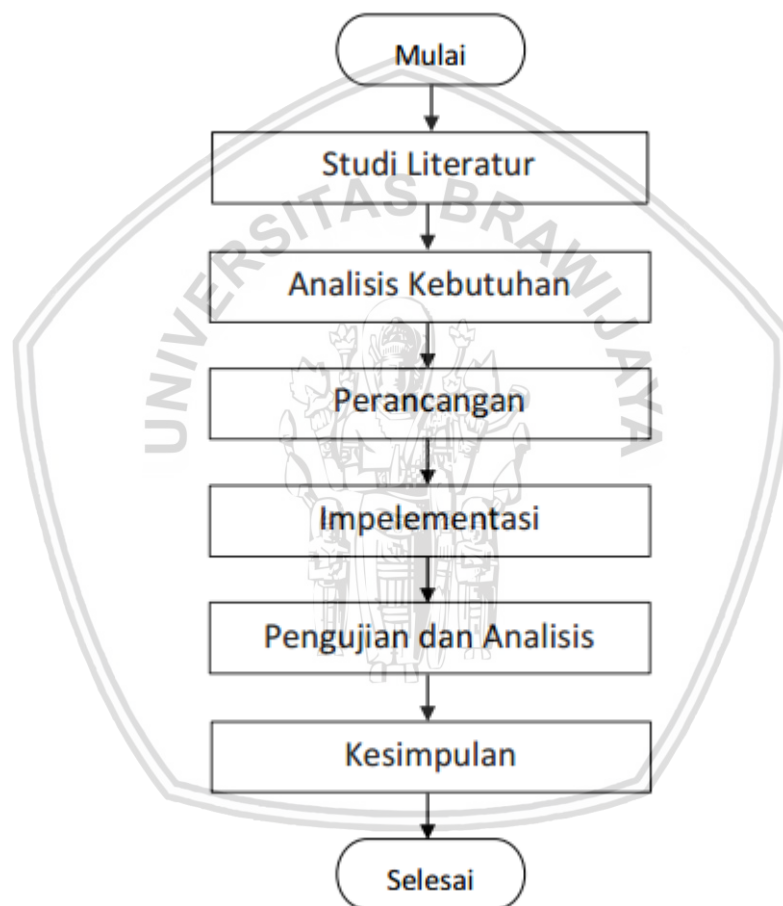
Menurut Kamus Besar Bahasa Indonesia (KBBI), sistem merupakan perangkat unsur yang secara teratur saling berkaitan sehingga membentuk suatu totalitas, sedangkan presensi memiliki arti kehadiran. Sistem presensi adalah sebuah perangkat atau cara untuk mengatur dan mengetahui kehadiran seseorang.

Sistem presensi biasanya digunakan untuk mengetahui tingkat kedisiplinan seseorang pada sebuah perusahaan atau instansi. Pada sebuah perusahaan atau instansi, sistem presensi biasanya dijadikan alat ukur untuk menghitung tingkat kehadiran seorang anggota atau pegawai.



## BAB 3 METODOLOGI

Bab ini berisi penjelasan mengenai metodologi atau langkah-langkah yang digunakan dalam penelitian. Penelitian ini merupakan pengembangan dari penelitian yang sudah ada dan bersifat implementatif. Penentuan alur metode penelitian sebagai langkah yang ditempuh untuk menyelesaikan penelitian ini secara sistematis. Gambar 3.1 merupakan diagram alir yang berisi tahapan-tahapan yang dilakukan dalam pembuatan sistem ini yang meliputi studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian dan analisis serta pengambilan keputusan.



Gambar 3.1 Diagram alir metodologi penelitian

### 3.1 Studi Literatur

Studi literatur merupakan tahapan pencarian literatur dan melakukan penyusunan teori dasar dan referensi yang akan menunjang penelitian ini. Studi literatur pada penelitian ini dilakukan dengan pemahaman terhadap tinjauan pustaka dan dasar teori yang dapat diperoleh dari buku, jurnal, skripsi dan website terkait. Tinjauan pustaka dan dasar teori tersebut meliputi:

1. *Fingerprint wifi*

*Fingerprint* wifi yang dimaksud disini adalah sebuah alamat MAC yang ditangkap oleh router ketika sebuah perangkat terkoneksi, alamat MAC bersifat unik sehingga dapat digunakan untuk mengidentifikasi sebuah perangkat yang terkoneksi pada router. Alamat MAC merupakan sebuah alamat fisik yang memiliki panjang 48 yang terletak pada lapisan *datalink* dari model OSI. Alamat MAC merupakan sebuah kode unik untuk setiap *device*.

## 2. API Untuk RouterOS

API menyediakan jaringan yang kaya fitur, salah satu fiturnya adalah dapat terintegrasi dengan sistem eksternal melalui API (*Application Programmable Interface*). Hal ini memungkinkan implementasi perangkat lunak pengguna untuk berkomunikasi dengan router agar mendapatkan informasi, konfigurasi dan pengelolaan perangkat tersebut.

## 3. Database

*Database* dapat diartikan sebagai kumpulan data (arsip) yang saling berhubungan yang disimpan dalam media penyimpanan elektronik dan terorganisir sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah untuk memenuhi kebutuhan. Mysql merupakan salah satu perangkat lunak yang digunakan untuk manajemen basis data.

## 4. Sistem Presensi

Sistem presensi adalah sebuah perangkat atau cara untuk mengatur dan mengetahui kehadiran seseorang.

### 3.2 Analisis Kebutuhan

Pada penelitian ini dilakukan analisis mengenai kebutuhan yang akan mendukung dalam proses penelitian. Analisa kebutuhan pada penelitian ini adalah sebagai berikut.

#### 3.2.1 Kebutuhan Fungsional

1. Fungsi sistem untuk mengambil data pada router atau *access point*.
2. Fungsi sistem untuk mendeteksi lokasi pegawai.
3. Fungsi sistem untuk membuat, mengambil, mengedit, menghapus, dan melihat *history* kehadiran pegawai pada *database*.

#### 3.2.2 Kebutuhan Perangkat Keras

1. Laptop : Digunakan untuk menjalankan sistem.
2. Mikrotik RB 951 Ui-2nD: Sebagai router yang terintegrasi langsung dengan sistem.
3. TP-Link WA801ND: Sebagai *access point* yang terhubung dengan router mikrotik.
4. D-Link DSL-2640B: Sebagai *access point* yang terhubung dengan router mikrotik.

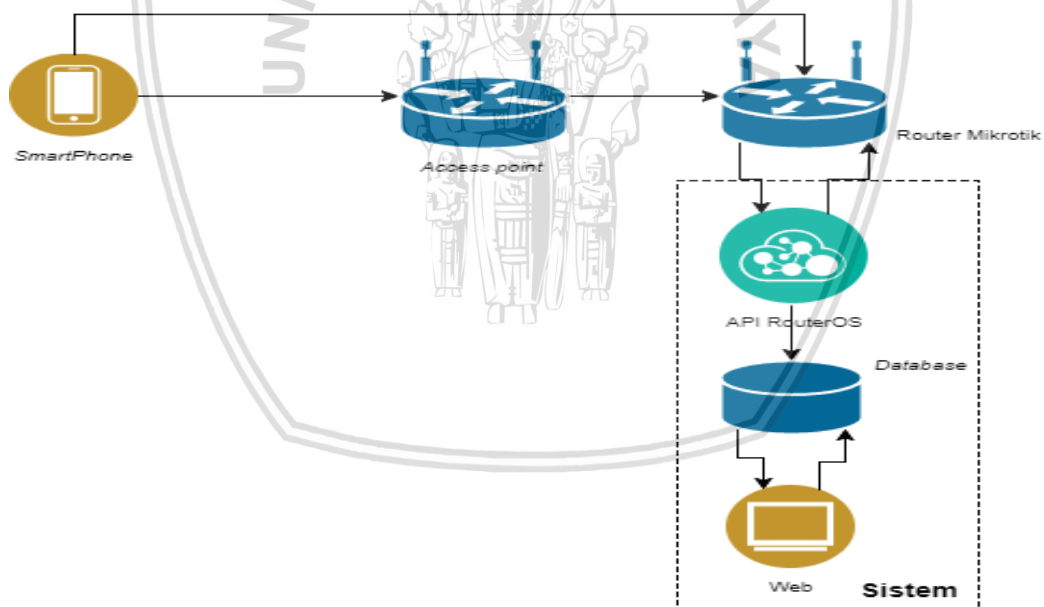
### 3.2.3 Perangkat Lunak

1. Windows 10 Pro 64-bit : Sistem operasi yang berjalan pada laptop yang digunakan untuk menjalankan sistem.
2. SublimeText2 : *Text editor* untuk berbagai jenis bahasa pemrograman salah satunya adalah PHP.
3. XAMPP : Sebagai localhost yang terdiri dari HTTP server, MYSQL database, dan penerjemahan bahasa yang ditulis menggunakan bahasa pemrograman PHP dan Perl.
4. Google Chrome : Untuk menjalankan sistem pada web browser.

## 3.3 Perancangan Sistem

Perancangan sistem dilakukan setelah melewati tahap analisis kebutuhan sistem yang harus terpenuhi terlebih dahulu baik dari sisi perangkat keras dan perangkat lunak. Tujuan perancangan sistem agar tahapan implementasi berjalan secara sistematis dan terstruktur.

### 3.3.1 Desain Alur Kerja Sistem



**Gambar 3.2 Desain alur kerja sistem**

Uraian setiap blok pada diagram seperti yang ditunjukkan pada Gambar 3.2 adalah sebagai berikut :

1. *Smartphone* akan melakukan koneksi melalui *access point* yang terintegrasi dengan router mikrotik, atau dapat langsung terkoneksi melalui router mikrotik, dikarenakan router yang digunakan merupakan *wireless router*.



2. Ketika *smartphone* melakukan koneksi ke router atau *access point*, router atau *access point* akan mengambil data-data yang diperlukan untuk router, salah satunya adalah alamat MAC, yang digunakan router untuk membedakan suatu perangkat dengan perangkat yang lainnya, jika perangkat terkoneksi pada *access point* maka *access point* juga akan memberikan informasi alamat MAC kepada router mikrotik.
3. Data alamat MAC yang tersimpan pada router kemudian akan diambil melalui API . Sebelum mengakses, API harus membuat koneksi terlebih dahulu ke router mikrotik, kemudian membaca data alamat MAC yang ada pada router mikrotik. Setelah itu alamat MAC yang dibaca akan dimasukkan ke *database*.
4. Di dalam *database* alamat MAC kemudian akan disimpan pada *database*. *Database* memiliki lebih dari satu tabel yang saling berelasi, dimana ada tabel yang akan menyimpan data milik pegawai, dan tabel yang menyimpan alamat MAC beserta waktu presensi. Data tersebut nantinya akan dibandingkan sesuai dengan data alamat MAC yang dimiliki pegawai.
5. Pegawai yang terdeteksi hadir oleh sistem akan ditampilkan pada halaman web.
6. Untuk menampilkan posisi pegawai API tidak perlu menyimpannya ke *database*, tetapi langsung menampilkan pada web admin.

### 3.4 Implementasi Sistem

Pada tahap ini hanya dijelaskan gambaran umum implementasi sistem, untuk lebih jelasnya akan dijelaskan pada bab selanjutnya yaitu bab implementasi sistem. Pertama yang harus dilakukan adalah memastikan semua rancangan diterapkan pada perangkat-perangkat yang telah disiapkan. Perangkat lunak yang dibutuhkan diinstall di perangkat keras yang tersedia. Laptop harus terinstall XAMPP dan Winbox untuk melakukan *konfigurasi* pada router, dan harus diuji apakah XAMPP dapat berjalan dengan baik dan Winbox dapat terkoneksi dengan baik pada router. *Access point* harus tersambung dengan benar pada router mikrotik sehingga jika ada perangkat yang tersambung pada *access point* maka router juga dapat membaca perangkat tersebut. Jika semua sudah berjalan dengan baik kemudian hidupkan router, jalankan sistem pada web browser, kemudian *smartphone* melakukan koneksi ke router atau *access point*, pastikan alamat MAC dan data-data pegawai sudah terdaftar pada *database*, jika sudah terkoneksi dan terdaftar, Siste, akan menampilkan data bahwa user telah melakukan presensi dan menampilkan dimana posisi pegawai tersebut.

### 3.5 Pengujian dan Analisis

Tahapan pengujian dan analisis akan dilakukan untuk mengetahui tingkat kinerja dari sistem presensi sudah sesuai dengan kebutuhan. Pengujian dikatakan berhasil jika hasil pengujian tidak menunjukkan kesalahan. Pengujian dibagi pada 2 garis besar yaitu :

### 1. Pengujian Fungsional

Pengujian fungsional merupakan pengujian secara *black-box* yang dilakukan untuk menguji fungsionalitas sistem apakah sudah berjalan dengan baik dan benar. Pengujian dilakukan dengan interaksi antara sistem pendeteksi kehadiran, router, dan *database*. Pengujian ini dianggap berhasil jika semua fungsi yang dipaparkan tidak menunjukkan adanya kegagalan dan kesalahan dalam pengujian.

### 2. Pengujian Performa

Pengujian performa dilakukan untuk menguji performa dari sistem. Pengujian berfokus pada waktu yang diperlukan sistem mendeteksi kehadiran pegawai, dimana dibagi menjadi 4 yaitu :

- a. Waktu yang diperlukan sistem terkoneksi pada router dan membaca alamat MAC dan *interface smartphone* yang ada pada router.
- b. Waktu yang diperlukan sistem untuk menampilkan data pegawai yang hadir atau yang diperlukan untuk mendeteksi perpindahan lokasi *smartphone* antara router atau *accesspoint*.
- c. Jumlah perangkat yang ditangani oleh sistem tiap detik.
- d. Waktu yang diperlukan sistem untuk mendeteksi kehadiran dari pertama terkoneksi ke router sampai dibaca oleh sistem. Pengujian akan dilakukan menggunakan wireshark.

## 3.6 Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan sudah dilakukan. Kesimpulan diambil berdasarkan data pengujian dari perangkat lunak yang dibangun. Pada tahap akhir dari penulisan ini terdapat saran yang digunakan untuk pengembangan sistem yang lebih baik dari sistem yang telah dibangun.

## BAB 4 ANALISIS KEBUTUHAN

### 4.1 Deskripsi Umum

#### 4.1.1 Perspektif Produk/Sistem

Sistem ini dapat berjalan dan bekerja apabila keseluruhan alat dan sistem telah dijalankan, dimulai dari *smartphone* yang digunakan untuk melakukan presensi, kemudian router mikrotik dan *access point* sebagai alat untuk *smartphone* terkoneksi dan mengambil alamat MAC dari *smartphone*, kemudian XAMPP sebagai *database* dan *localhost*, kemudian sistem yang harus selalu terkoneksi ke router. Sistem ini dapat mendeteksi kehadiran pegawai jika *smartphone* terkoneksi ke sebuah router atau *access point* yang telah ditentukan, kemudian alamat MAC yang telah dibaca oleh router atau *access point* akan ditangkap oleh sistem yang telah dibuat, dan dimasukkan ke dalam tabel pada *database*, tabel tersebut direlasikan dengan tabel lain untuk menghasilkan data kehadiran pegawai berupa nama, NIK, tanggal presensi, loker, dan lain-lain. Sistem juga mampu mendeteksi lokasi pegawai melalui router atau *access point* tempat terkoneksi *smartphone*.

#### 4.1.2 Kegunaan

Dengan adanya sistem ini *wifi* tidak hanya digunakan untuk bertukar data secara nirkabel tetapi juga dimanfaatkan untuk mendeteksi kehadiran seorang pegawai. Sistem ini tidak perlu menginstall aplikasi tertentu pada perangkat pegawai, pegawai cukup melakukan koneksi ke SSID (*Service Set Identifier*) router atau *access point* untuk melakukan presensi.

#### 4.1.3 Karakteristik Pengguna

Pengguna dalam implementasi sistem hanya berinteraksi dengan router atau *access point* menggunakan *smartphone*, dimana pegawai melakukan koneksi ke router atau *access point* yang telah ditentukan, setelah terkoneksi, router atau *access point* akan membaca alamat MAC dan *interface* tempat *smartphone* terkoneksi.

#### 4.1.4 Lingkungan Operasi

Sistem ini akan diterapkan pada gedung yang terdapat router dan *access point*. Sistem akan bekerja jika ada perangkat yang terkoneksi ke sebuah router atau *access point* yang telah ditentukan. Sistem bekerja pada sebuah wilayah yang memiliki lebih dari 1 (satu) gedung, atau pada gedung bertingkat, dimana sistem dapat bekerja dengan lebih dari 1 (satu) *access point* atau router.

#### 4.1.5 Batasan Perancangan dan Implementasi

Beberapa batasan yang telah diringkas dijelaskan sebagai berikut.

1. Sistem akan bekerja apabila router atau *access point* telah dihidupkan.

2. Sistem akan bekerja jika XAMPP telah dijalankan.
3. Sistem akan bekerja jika program dijalankan dan terkoneksi ke router.
4. Sistem akan bekerja jika ada perangkat yang terkoneksi pada router atau *access point* yang telah ditentukan.
5. Pada percobaan kali ini sistem hanya menggunakan 1 router dan 2 *access point*.
6. *Access point* terkoneksi pada router melalui port Ethernet.
7. Alamat MAC pada perangkat pegawai harus tersimpan terlebih dahulu pada *database*.

#### 4.1.6 Asumsi dan Ketergantungan

Beberapa asumsi dan ketergantungan sebagai persyaratan sistem adalah sebagai berikut :

1. Sistem hanya dapat mendeteksi kehadiran jika perangkat terkoneksi pada router atau *access point* yang telah ditentukan, dan alamat MAC pada perangkat tersebut harus sudah tersimpan di *database* sebelumnya.
2. Sistem hanya dapat mendeteksi lokasi pegawai dengan benar, jika *smartphone* tersebut dibawa oleh pegawai dan terkoneksi pada router atau *access point* yang telah ditentukan.

### 4.2 Kebutuhan Antarmuka Eksternal

#### 4.2.1 Antarmuka Pengguna

Pengguna yaitu pegawai cukup melakukan koneksi ke router atau *access point* melalui SSID (*Service Set Identifier*) yang telah ditentukan, kemudian router atau *access point* akan menangkap alamat MAC dan diberikan pada sistem untuk diproses.

#### 4.2.2 Antarmuka Perangkat Keras

1. Pengguna
  - a. *Smartphone* (yang dapat terhubung dengan router atau *access point*).
2. Sistem
  - a. Router Mikrotik RB 951 Ui-2<sup>nd</sup>
  - b. TP-Link WA801ND
  - c. D-Link DSL-2640B

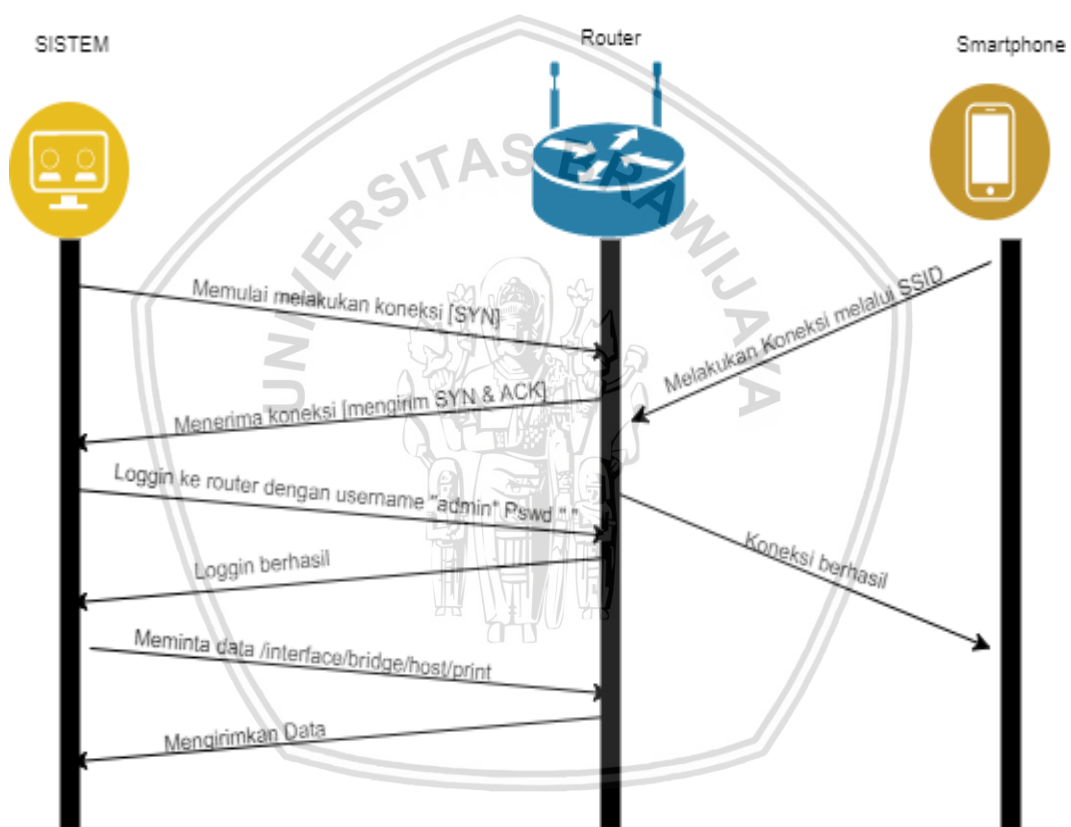
#### 4.2.3 Antarmuka Perangkat Lunak

Antar muka perangkat lunak untuk menjalankan sistem sebagai berikut:

1. *Database* dan WebServer

- a. Xampp
2. Text Editor
  - a. SublimeText2
3. Sistem Operasi
  - a. Windows 10 Pro 64-bit
4. WebBrowser
  - a. Google Chrome.

#### 4.2.4 Antarmuka Komunikasi



**Gambar 4.1** Antar muka komunikasi

Komunikasi yang terjadi pada sistem yang pertama *smartphone* melakukan koneksi ke router, setelah terkoneksi sistem akan melakukan pengecekan dan melakukan koneksi terhadap router. Setelah terkoneksi, sistem akan melakukan *login* ke router untuk membaca siapa yang terkoneksi pada router.

#### 4.3 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus dipenuhi agar suatu sistem dapat bekerja sesuai keinginan, beberapa kebutuhan fungsional yang harus ada pada sistem dijelaskan sebagai berikut.

Tabel 4.1 Kebutuhan fungsional

Kode	Fungsi	Keterangan
PF_001	Fungsi sistem melakukan koneksi ke router.	Sistem dapat terkoneksi pada router yang telah ditentukan.
PF_002	Fungsi sistem mendeteksi kehadiran pegawai.	Sistem dapat mendeteksi kehadiran pegawai melalui alamat MAC pada <i>smartphone</i> pegawai.
PF_003	Fungsi sistem mendeteksi lokasi pegawai yang hadir.	Sistem dapat mendeteksi lokasi pegawai berdasarkan tempat <i>smartphone</i> pegawai terkoneksi.
PF_004	Fungsi sistem menambahkan data pegawai ke <i>database</i> .	Sistem dapat menambahkan data pegawai berupa nama, NIK, Divisi, dan alamat MAC <i>smartphone</i> .
PF_005	Fungsi sistem menghapus data pegawai pada <i>database</i> .	Sistem dapat menghapus data pegawai pada <i>database</i> .
PF_006	Fungsi sistem melakukan <i>edit</i> data pegawai.	Sistem dapat melakukan <i>edit</i> data pegawai pada <i>database</i> .
PF_007	Fungsi sistem menampilkan daftar data pegawai yang ada pada <i>database</i> .	Sistem dapat menampilkan seluruh data pegawai yang ada pada <i>database</i> .
PF_008	Fungsi sistem menampilkan <i>history</i> kehadiran pegawai sesuai tanggal atau nama yang ingin ditampilkan.	Sistem dapat menampilkan <i>history</i> kehadiran pegawai sesuai tanggal atau nama yang ingin ditampilkan.
PF_009	Fungsi sistem melakukan insert data berupa alamat MAC ke dalam <i>database</i> .	Sistem dapat langsung menyimpan data alamat MAC <i>smartphone</i> pegawai yang hadir pada <i>database</i> .

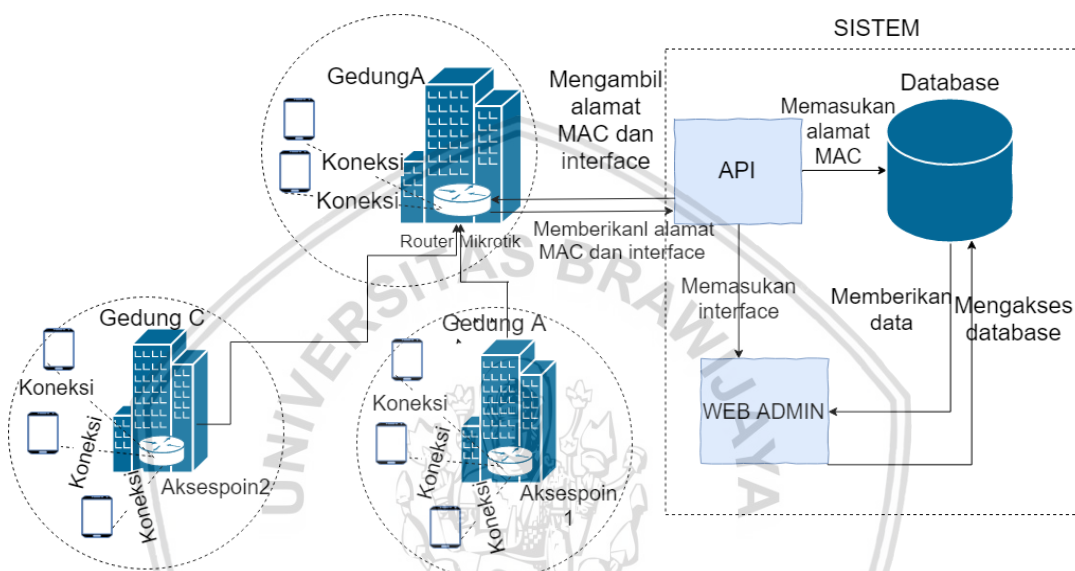


## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan tentang perancangan sistem yang dilakukan pada penelitian ini, serta implementasi sistem.

### 5.1 Perancangan Sistem

#### 5.1.1 Gambaran Umum Sistem

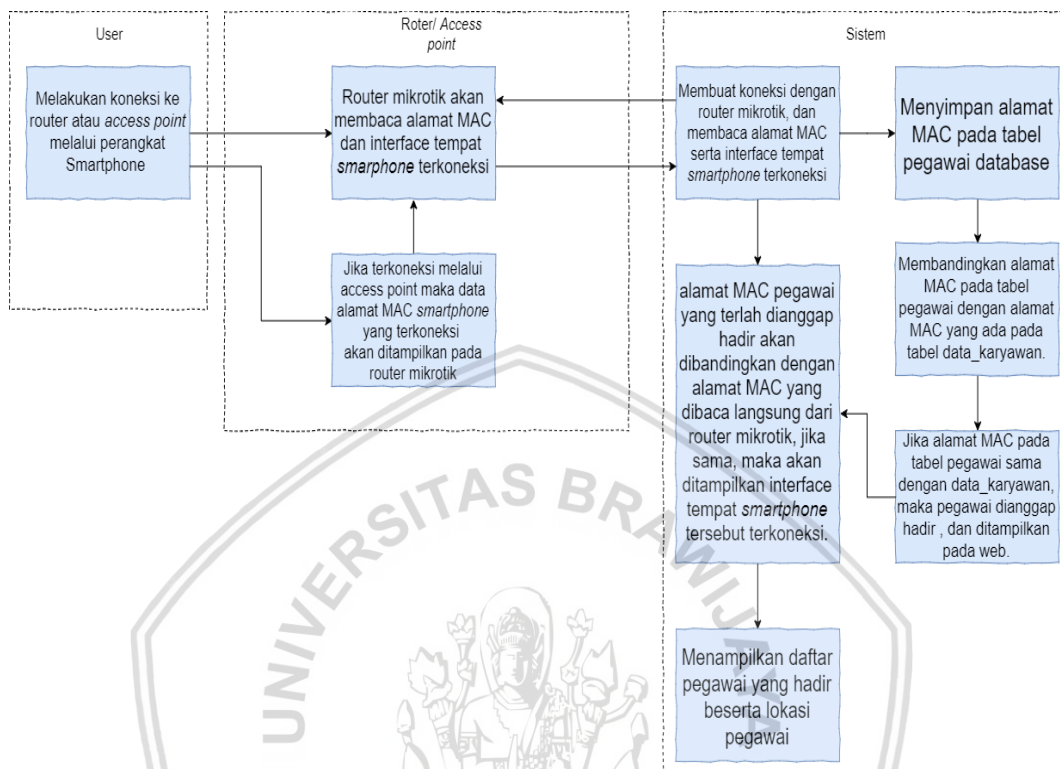


**Gambar 5.1 Konsep umum Sistem Pendeteksi Kehadiran**

Gambar 5.1 menjelaskan tentang konsep umum arsitektur sistem kehadiran pegawai yang akan dibuat. Router pada Gedung A adalah router mikrotik, *smartphone* dapat terkoneksi pada router pada Gedung A karena router mikrotik tersebut merupakan *wireless* router, sedangkan router yang ada pada Gedung B dan Gedung C dijadikan sebagai *access point* yang terhubung ke router mikrotik melalui *port* ethernet, sehingga ketika ada *smartphone* yang terkoneksi ke *access point* Gedung B dan Gedung C maka alamat MAC beserta *interface* tempat *smartphone* terkoneksi akan terbaca router mikrotik yang ada pada Gedung A, jika perangkat terkoneksi router Gedung A maka *interface* yang akan muncul adalah di Gedung A begitu pula sebaliknya, parameter inilah yang digunakan untuk menentukan posisi pegawai. Agar pegawai dapat melakukan presensi, yang harus dilakukan adalah *smartphone* melakukan koneksi ke router mikrotik atau *access point* yang telah ditentukan, setelah terkoneksi, router mikrotik akan menyimpan alamat MAC dari *smartphone* yang terkoneksi pada router mikrotik atau *access point* dimana perangkat tersebut terkoneksi (melalui *interface* dimana router atau *access point* tersebut terhubung). Kemudian API mengambil alamat MAC pada router mikrotik, dan menyimpannya pada *database*, untuk *interface*, API tidak menyimpannya ke *database*, melainkan langsung

ditampilkan pada web, karena *interface* disini digunakan sebagai parameter untuk menentukan lokasi pegawai tersebut berada dan dapat berubah-ubah.

### 5.1.2 Alur Kerja Sistem

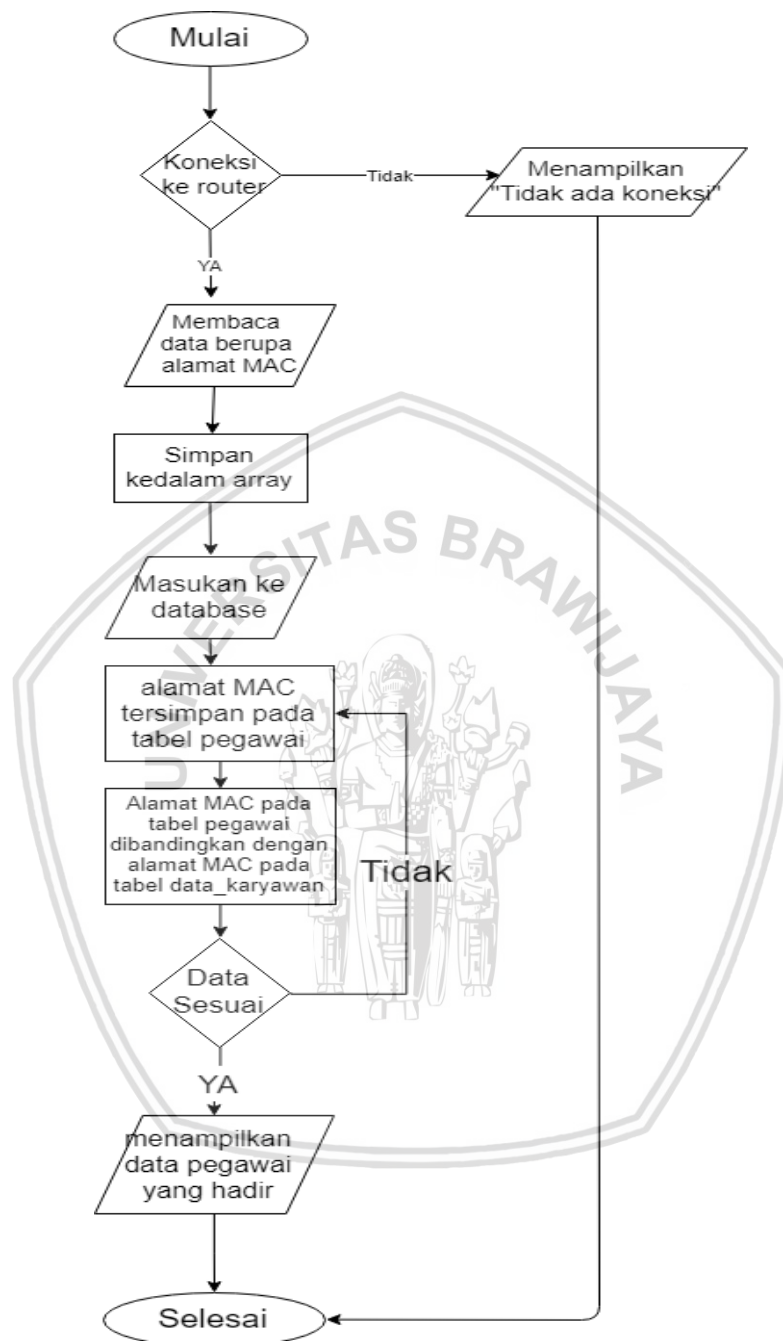


Gambar 5.2 Alur Kerja Sistem

Pada Gambar 5.2, dijelaskan tentang alur kerja sistem, dimulai dari pegawai melakukan koneksi menggunakan *smartphone* ke router atau *access point* untuk melakukan presensi. Alamat MAC serta *interface* tempat *smartphone* terkoneksi akan dibaca oleh router mikrotik seperti Gambar 5.1. Untuk menentukan kehadiran seorang pegawai sistem melihat dari alamat MAC yang masuk pada tabel pegawai kemudian dibandingkan dengan alamat MAC yang tersimpan pada tabel data\_karyawan, jika alamat MAC sama maka pegawai akan dianggap hadir dan ditampilkan pada web daftar hadir pegawai, untuk mengetahui lokasi pegawai yang hadir, alamat MAC pegawai yang hadir akan dibandingkan dengan alamat MAC yang dibaca langsung oleh dari router, jika sama maka lokasi pegawai yang hadir akan ditampilkan.

### 5.1.3 Perancangan Perangkat Lunak

#### A. Mendeteksi kehadiran pegawai

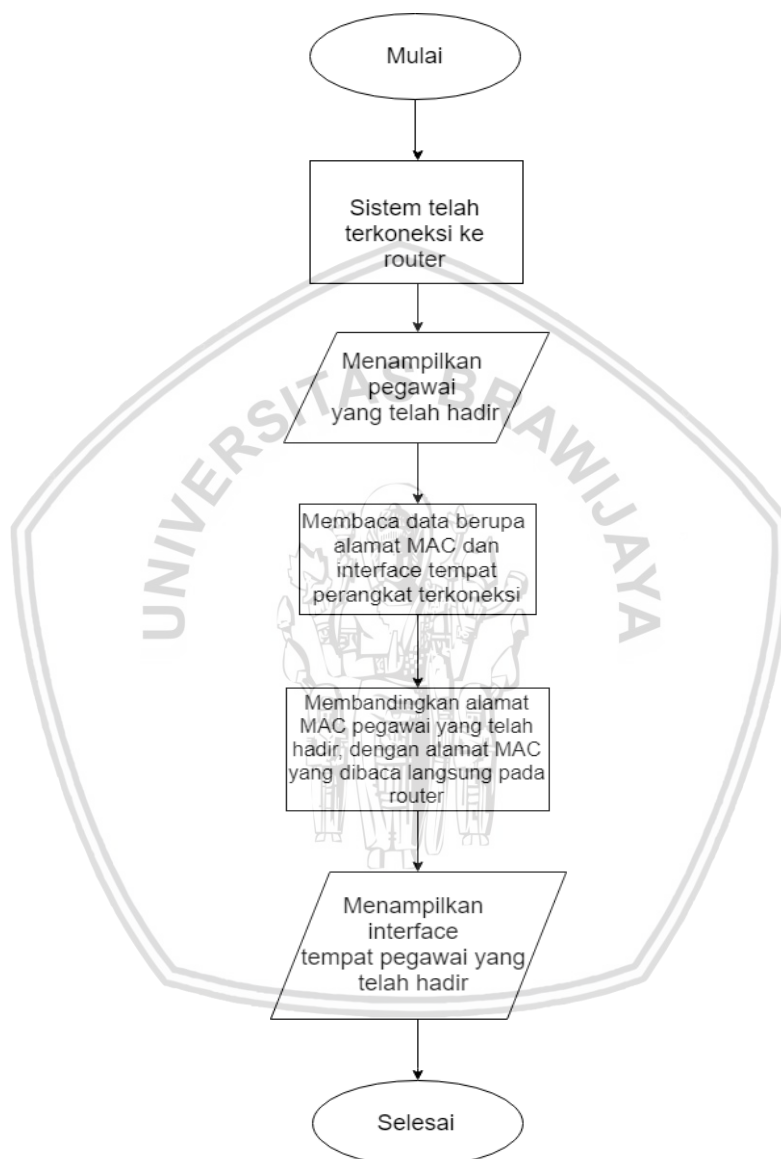


**Gambar 5.3 Mendeteksi kehadiran**

Gambar 5.3 menjelaskan tentang alur kerja sistem untuk mendeteksi kehadiran pegawai. Pada saat sistem dijalankan, sistem akan melakukan koneksi ke router melalui alamat IP yang telah ditentukan, jika tidak terkoneksi maka akan muncul pesan “Tidak ada koneksi”, jika terkoneksi maka sistem akan membaca alamat MAC perangkat yang terkoneksi pada router atau *access point*, kemudian data tersebut disimpan ke *database*. Untuk menentukan kehadiran seorang

pegawai sistem melihat dari alamat MAC yang masuk pada tabel pegawai kemudian dibandingkan dengan alamat MAC yang tersimpan pada tabel data\_karyawan, jika alamat MAC sama, maka pegawai akan dianggap hadir dan ditampilkan pada web daftar hadir pegawai. Jika tidak sama maka tidak akan ditampilkan. Sistem akan melakukan pengecekan pada router setiap 5 detik.

#### B. Mendeteksi lokasi pegawai

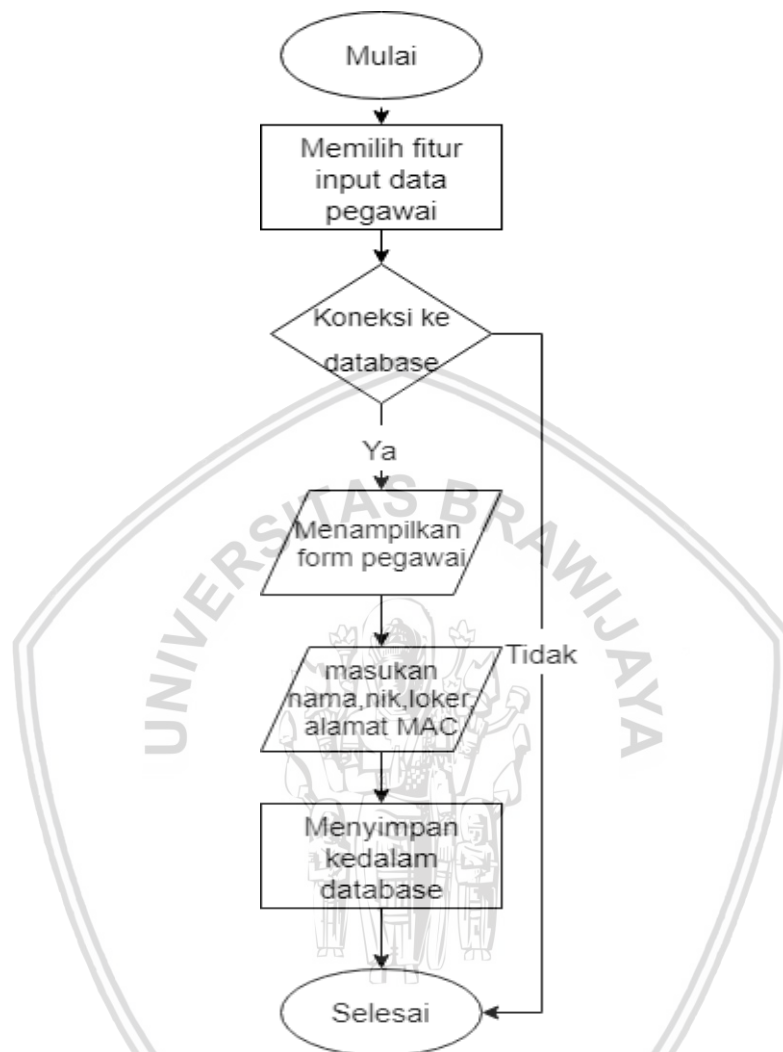


**Gambar 5.4 Mendeteksi lokasi**

Pada Gambar 5.4 digambarkan alur kerja sistem untuk mendeteksi lokasi pegawai. Saat sistem telah mendeteksi pegawai yang hadir, maka alamat MAC dari pegawai yang hadir akan dibandingkan dengan alamat MAC yang dibaca langsung dari router mikrotik, jika alamat MAC sama, maka *interface* tempat *smartphone* terkoneksi akan ditampilkan. Ada 3 *interface* pada penelitian ini, yang pertama Gedung A, sistem akan menampilkan Gedung A jika *smartphone* terkoneksi pada router mikrotik, kedua Gedung B, sistem akan menampilkan Gedung B jika

*smartphone* terkoneksi pada *access point* 1, begitu pula jika terkoneksi ke *access point* 2 maka akan menampilkan Gedung C.

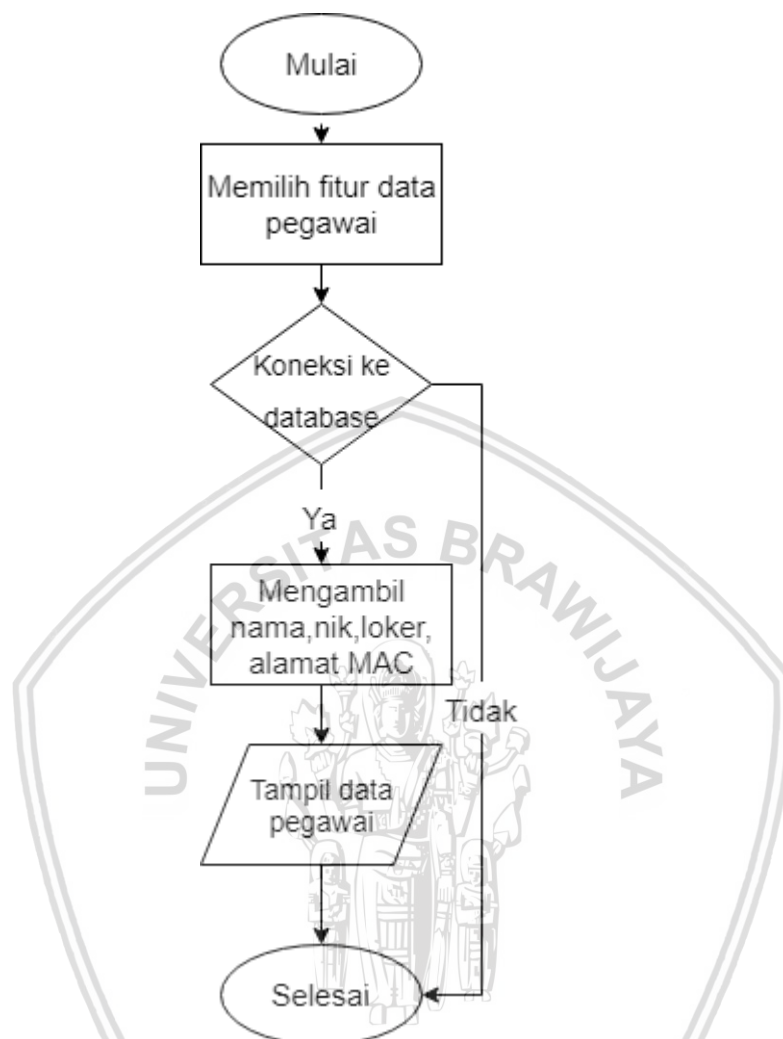
C. Menambahkan data pegawai



**Gambar 5.5 Alur kerja menambahkan pegawai**

Pada Gambar 5.5 digambarkan alur kerja sistem untuk menambahkan data pegawai. Pertama memilih fitur *input* data pegawai, melakukan koneksi ke *database*, lalu *form* input akan ditampilkan, kemudian data pegawai diinputkan ke dalam *form* data pegawai setelah diinputkan maka data pegawai diproses untuk disimpan dalam *database*.

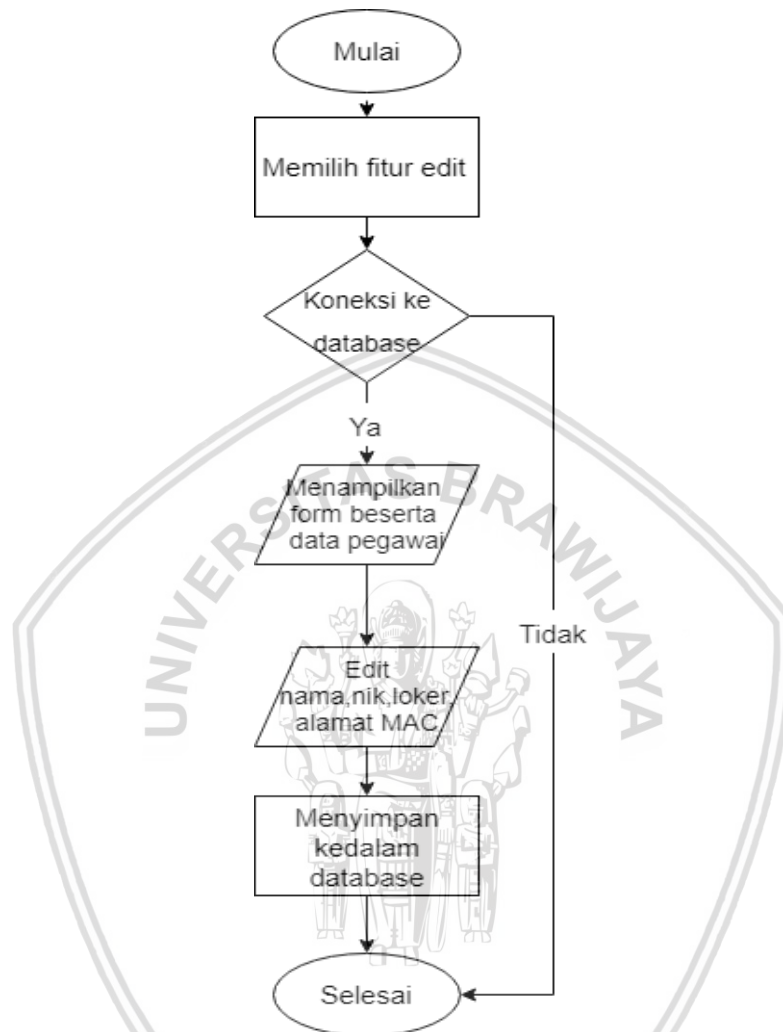
## D. Menampilkan data pegawai

**Gambar 5.6 Alur kerja menampilkan data pegawai**

Pada Gambar 5.6 digambarkan alur kerja sistem untuk menampilkan data pegawai. Pertama memilih fitur data pegawai, kemudian melakukan koneksi ke *database*, lalu mengambil data pegawai pada *database*, setelah itu ditampilkan pada halaman web daftar pegawai.



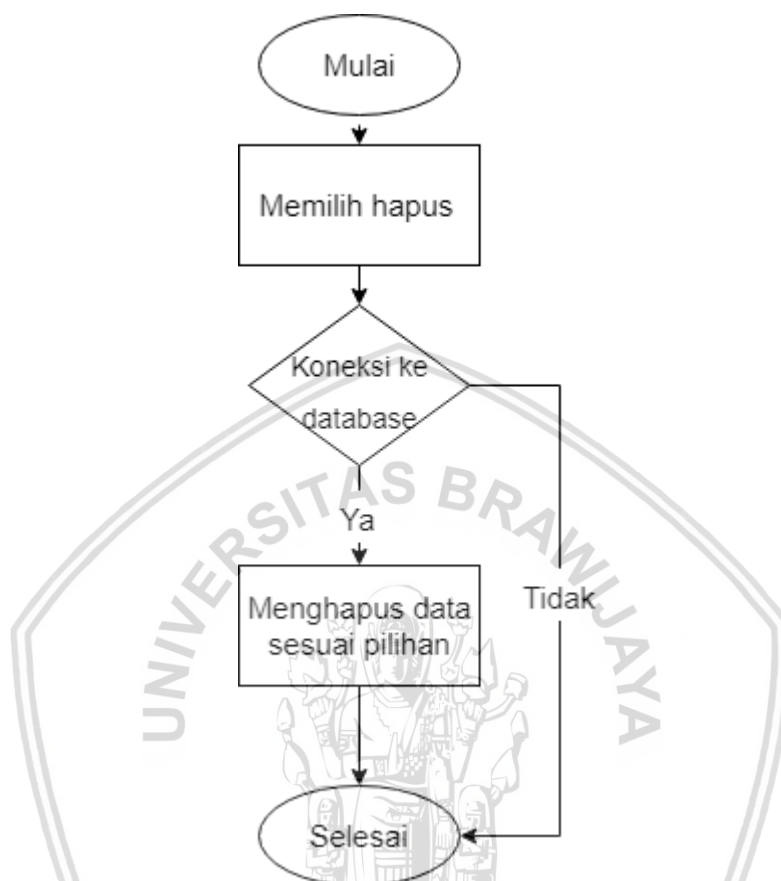
E. Edit data pegawai



**Gambar 5.7 Alur kerja edit data pegawai**

Pada Gambar 5.7 digambarkan alur kerja sistem untuk mengedit data pegawai. Pertama memilih fitur edit, kemudian sistem melakukan koneksi ke *database*, lalu menampilkan form edit pegawai beserta data pegawai pada *database*, data yang telah diambil kemudian diedit dan disimpan ke dalam *database*.

## F. Hapus data pegawai

**Gambar 5.8 Alur kerja hapus data pegawai**

Pada Gambar 5.8 digambarkan alur kerja sistem untuk menghapus data pegawai. Pertama memilih hapus pada data pegawai yang akan dihapus, kemudian melakukan koneksi ke *database*, jika terkoneksi maka data pegawai yang dipilih akan dihapus dalam *database*.

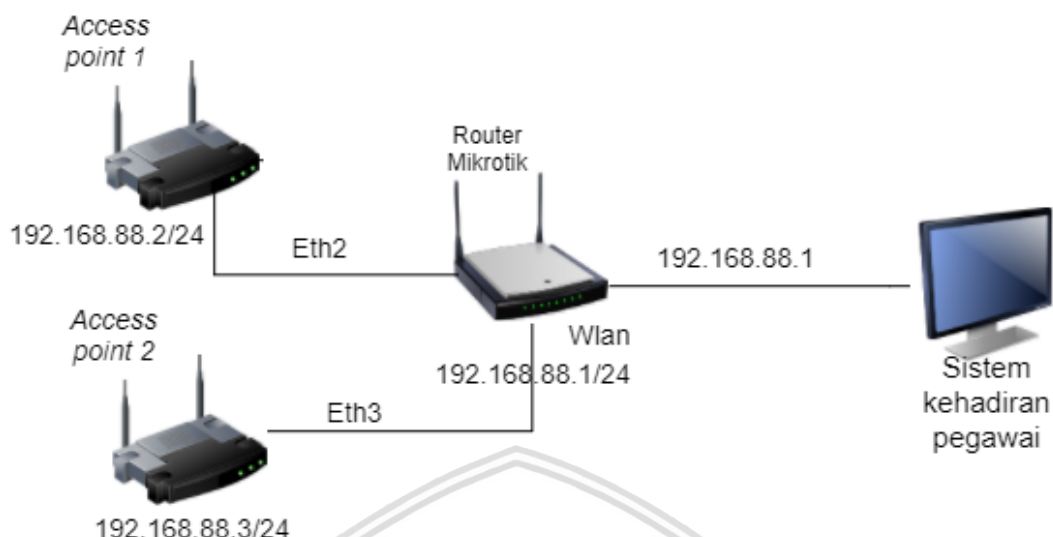
G. Menampilkan *history* pegawai yang hadir sesuai dengan tanggal kehadiran.



**Gambar 5.9** Alur kerja menampilkan *history* berdasarkan tanggal dan nama

Pada Gambar 5.9 dijelaskan alur kerja sistem menampilkan *history* pegawai berdasarkan nama. Pertama saat memilih menu *history* maka sistem akan langsung menampilkan seluruh data pegawai yang pernah hadir, kemudian akan ada pilihan “Menampilkan Berdasarkan Nama”, jika melakukan pilihan tersebut maka akan menampilkan halaman baru berisi *form* pencarian beserta daftar pegawai yang pernah hadir, selanjutnya masukan nama yang ingin ditampilkan pada *form* pencarian, kemudian klik submit, halaman akan menampilkan daftar pegawai yang pernah hadir berdasarkan nama yang dimasukan pada *form*. Untuk menampilkan data pegawai yang hadir pada tanggal tertentu maka harus memilih tanggal pada *form* tanggal, setelah memilih tanggal, halaman akan menampilkan sesuai dengan tanggal yang dipilih.

#### 5.1.4 Perancangan Topologi Jaringan



**Gambar 5.10 Perancangan topologi jaringan**

Pada Gambar 5.10 terlihat bahwa router mikrotik terhubung dengan *access point* dengan cara di *bridge* melalui *port ethernet*, sehingga jika ada perangkat yang terkoneksi pada *access point* maka data perangkat yang terkoneksi tersebut juga dapat dilihat oleh router mikrotik. Router mikrotik yang digunakan pada penelitian ini adalah wireless router sehingga perangkat juga bisa langsung terkoneksi pada router mikrotik pada alamat IP 192.168.88.1/24. Semua alamat MAC dan *interface* yang akan diproses oleh sistem diambil pada router mikrotik dengan cara terkoneksi terlebih dahulu dengan router mikrotik melalui alamat IP 192.168.88.1.

#### 5.1.5 Perancangan Database

Perancangan *database* adalah untuk menampung data yang dilakukan secara terstruktur, sehingga data dapat dibaca dan dipahami dengan mudah. Pada penelitian ini memiliki dua table utama yaitu table pegawai dan data karyawan.

##### 1. Tabel data\_karyawan

**Tabel 5.1 Data karyawan**

Nama Attribute	Type	Null
nama	Varchar (50)	No
nik	Varchar (50)	No
loker	Varchar (50)	No
mac2	Varchar (50)	No

Pada Tabel 5.1 data\_karyawan digunakan untuk menyimpan data-data pegawai seperti nama, nik, loker, dan mac2. Mac2 pada Tabel 5.1 merupakan *primary key* yang bersifat unik untuk membedakan data satu dengan yang lainnya, agar sistem ini dapat mendeteksi kehadiran, seorang pegawai harus mengisi

terlebih dahulu data pegawai yang diperlukan pada data\_karyawan, tabel ini digunakan sebagai parameter untuk membedakan antara pegawai dengan pegawai yang lainnya.

## 2. Tabel pegawai

**Tabel 5.2 Data pegawai**

Nama <i>Attribute</i>	Type	Null
id	Varchar (50)	No
mac	Varchar (50)	No
date	Date : datetime	No

Tabel 5.2 Data pegawai digunakan untuk menyimpan alamat MAC pada perangkat yang terkoneksi pada router atau *access point*. Ketika menyimpan alamat MAC pada *database*, program menyimpan tanggal dan waktu saat alamat MAC tersebut dimasukan ke dalam *database* untuk mengetahui waktu seorang pegawai melakukan presensi. Tabel pegawai saling berelasi dengan tabel data\_karyawan untuk menampilkan siapa saja pegawai yang telah hadir, caranya adalah dengan membandingkan alamat MAC yang masuk pada tabel pegawai dengan alamat MAC pada tabel data\_karyawan, jika sama maka sistem akan menampilkan nama, nik, dan loker pegawai yang hadir. *Attribute* id pada Tabel 5.2 digunakan untuk membedakan antara alamat MAC yang masuk pada database, sehingga ketika ada alamat MAC yang sama maka akan tetap terbaca oleh sistem, hal ini berguna ketika pegawai melakukan presensi pada hari berikutnya.

## 5.2 Implementasi Sistem

### 5.2.1 Mendeteksi Kehadiran Pegawai

Untuk mengambil alamat MAC perangkat yang terkoneksi pada router maka sistem harus terkoneksi terlebih dahulu ke router.

**Tabel 5.3 Contoller koneksi ke router**

```
<?php
require ('RouterOS_api.php');
public function koneksi ()
{
    $API = new RouterOSAPI ();
    $API->debug = false;
    if ($API->connect ('192.168.88.1', 'admin', '')){
```

Pada Tabel 5.3 berisi program yang memanggil fungsi pada *library* yaitu RouterOS\_api.php (Lampiran 2). Agar program sistem dapat terkoneksi ke router, maka terlebih dahulu harus memasukan *library* ke dalam program sistem yang dibuat, kemudian menginstansiasi RouterOSAPI (), ini adalah class dari RouterOS\_api.php yang merupakan *library* dari sistem yang dibuat pada penelitian

kali ini. Untuk melakukan koneksi ke router maka memanggil fungsi `connect` pada class `RouterOSAPI ()`, kemudian mengatur alamat IP sesuai dengan alamat IP pada router mikrotik yang ingin diambil datanya.

**Tabel 5.4 Controller insert alamat MAC**

```
public function koneksi ()
{
    $API = new RouterOSAPI ();
    $API->debug = false;
    if ($API->connect ('192.168.88.1','admin','')){
        $ARRAY = $API->write ('/interface/bridge/host/print');
        $ARRAY = $API->read ();
        include "condatabase.php";
        foreach ($ARRAY as $value) {
            $mac= $value ['mac-address'];
            date_default_timezone_set ('Asia/Jakarta');
            $date=date ("Y-m-d");
            $cek = mysqli_query ($koneksi, "SELECT mac FROM
pegawai WHERE mac ='$mac' AND date = '$date'");

            $cek_rows = mysqli_num_rows ($cek);

            if ($cek_rows==0) {

                $tambah=mysqli_query ($koneksi,"INSERT INTO
pegawai VALUES ('null','$mac','$date')");
            } else {
            }
        }
        $API->disconnect ();

    }else {
        echo " tidakbisa";
    }
    return $ARRAY;
}
```

Setelah terkoneksi dengan router, kemudian memanggil fungsi `write` dan `read` untuk mengambil data alamat MAC yaitu `$value ['mac-address']` pada router, alamat MAC tersebut di simpan dalam array terlebih dahulu kemudian di *insert* ke dalam *database*. Seperti pada Tabel 5.4.



**Tabel 5.5 Model menampilkan kehadiran**

```

public function getkehadiran ()
{
    include "condatabase.php";
    $data_karyawan = array ();

    $query = mysqli_query ($koneksi, "SELECT
data_karyawan.nama,data_karyawan.nik,data_karyawan.loker,pegawai
.mac,pegawai.date from pegawai, data_karyawan where pegawai.mac=
data_karyawan.mac2 order by pegawai.date desc ") or die
(mysqli_error ());

    if (mysqli_num_rows ($query) == 0){
        echo "<b>Tidak ada data yang tersedia</b>";
    }else{

        while ($r = $query->fetch_assoc ()){
            $data_karyawan[] = $r;
        }
    }return $data_karyawan;
}

```

Pada Tabel 5.5 merupakan program untuk mendeteksi kehadiran pegawai. Alamat MAC yang masuk pada tabel pegawai harus sama dengan alamat MAC pada data\_karyawan. Pada *sourcecode* \$query Tabel 5.5, melakukan *select* pada tabel pegawai, dan tabel data\_karyawan, pada proses ini pendeteksian kehadiran dilakukan, jika alamat MAC yang ada pada tabel pegawai sama dengan yang ada pada tabel data\_karyawan, maka pegawai di anggap hadir.

**Tabel 5.6 Controller menampilkan kehadiran**

```

public function tampil ()
{
    $data_karyawan=$this->model->getkehadiran ();
    $ARRAY = $this->koneksi ();
    include 'View/View.php';}

```

Pada tabel 5.6 merupakan sebuah controller yang menghubungkan antara database dan sistem. Fungsi *tampil()* digunakan untuk menampilkan pegawai yang hadir pada halaman View.php, dimana data tersebut dibaca dari *database* melalui fungsi *getkehadiran()*.

**Tabel 5.7 View halaman kehadiran**

```

<!DOCTYPE html>
<html>
<head>

```

```

<title>PRESENSI SKJ</title>
<meta http-equiv="refresh"content="5">
</head>
<body>
<br>
<center><h1 class="mt-5">Presensi</h1><br>

<table border="1" cellspacing="0" cellpadding="4">
  <tr style="text-align:center;background-color:#ff8080">
    <td> No </td>
    <td>Nama</td>
    <td>NIK</td>
    <td>Loker</td>
    <td>Lokasi</td>
    <td>Tanggal</td>
  </tr>

  <?php
    $no=1;
    foreach ($data_karyawan as $r){ ?>

    <tr style="text-align:center">
      <td><?php echo "$no" ?> </td>
      <td><?php echo $r['nama'] ?></td>
      <td><?php echo $r['nik'] ?></td>
      <td><?php echo $r['loker'] ?></td>
      <td>
    </td>
      <td><?php echo $r['date'] ?></td>
    </tr>

    <?php
    $no++;

    }
  ?>
</table>
</form>
<a href ="delete.php"><button>delete</button>&nbsp;

```

```

</center>
</body>
</html>

```

Jika alamat MAC pada tabel pegawai sama dengan alamat MAC yang ada pada tabel data\_karyawan, maka sistem akan menampilkan nama, nik, loker, dan waktu saat pegawai tersebut dianggap hadir pada halaman View.php, seperti pada Tabel 5.7.

### 5.2.2 Mendeteksi Lokasi Pegawai

**Tabel 5.8 Contoller mendeteksi lokasi**

```

<?php
include_once ('model/model.php');
require ('RouterOS_api.php');
class controller{
    public $model;

    public function __construct (){
        $this->model = new model ();
    }

    public function koneksi ()
    {

        $API = new RouterOSAPI ();
        $API->debug = false;
        if ($API->connect ('192.168.88.1','admin','')){
            $ARRAY = $API->write ('/interface/bridge/host/print');
            $ARRAY = $API->read ();}}

```

Untuk mendeteksi lokasi pegawai hampir sama dengan mendeteksi kehadiran, yang membedakan adalah untuk mendeteksi lokasi, data tidak perlu dimasukan terlebih dahulu ke *database*. Sistem langsung membaca alamat MAC dan *interface* melalui router. Sistem akan mengambil data *interface/bridge/host* Seperti pada Tabel 5.8.

**Tabel 5.9 Deteksi lokasi pegawai**

```

<td>Lokasi</td>
<td><?php
foreach ($ARRAY as $value){?>
    <?php
    if ($r['mac']==$value['mac-address']){

```

```

echo $value['on-interface'];}}?>

<?php
    }?></td>

```

Terlihat pada Tabel 5.9, untuk mendeteksi lokasi program ini menggunakan *interface* yaitu `$value['on-interface']` yang menunjukan pada router atau *access point* mana *smartphone* terkoneksi. Alamat MAC yang langsung dari router yaitu `$value['mac-address']` dengan alamat MAC yang ada pada *database* yaitu `$r['mac']` akan dicocokkan untuk mengenali lokasi pegawai yang hadir tersebut berada.

### 5.2.3 Menambah Data Pegawai

**Tabel 5.10 View form input data pegawai**

```

<form action="" method="POST">
<table>
<tr>
<td>Nama</td>
<td><input type="text" name="nama" placeholder="Nama" /></td>
</tr>
<tr>
<td>NIK </td>
<td><input type="text" name="nik" placeholder="NIK" /></td>
</tr>
<tr>
<td>Loker</td>
<td><input type="text" name="loker" placeholder="Loker"></td>
</tr>
<tr>
<td>Mac</td>
<td><input type="text" name="mac2" placeholder="mac"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" name="tambah" value="Tambah">
<button type="reset" value="Reset">Reset</button>
</td>
</tr>
</table>
</form>

```

Untuk memasukan data pegawai, yang pertama adalah membuat *form* terlebih dahulu seperti Tabel 5.10, untuk mengirimkan data yang ingin dimasukan

ke dalam *database*. *Form* merupakan method “*Post*” yang berisi nama, nik, mac2, dan loker.

**Tabel 5.11 Model menambahkan pegawai**

```
public function addpegawai ()
{
include "condatabase.php";
if (isset ($_POST['tambah'])) {
    echo isset ($_POST['tambah']);
    $nama      = $_POST['nama'];
    $nik       = $_POST['nik'];
    $loker     = $_POST['loker'];
    $mac2      = $_POST['mac2'];

    $querytambah = mysqli_query ($koneksi, "INSERT INTO
data_karyawan VALUES ('$nama', '$nik', '$loker', '$mac2')") or
die (mysqli_error ());
    if ($querytambah) {
        header ('');
    } else {
        echo "Upss Something wrong..";
    }
}
}
```

Setelah itu dilakukan pengecekan pada `($_POST['tambah'])`, jika terdapat data yang dikirim maka setelah itu membuat *variable* baru untuk mengambil data-data yang dikirim seperti `$_POST['nama']`, `$_POST['nik']`, `$_POST['loker']`, `$mac2 = $_POST['mac2']`, dalam `($_POST['tambah'])` yang bisa dilihat pada Tabel 5.11. Setelah itu data dalam *variable* diinsert ke dalam *database*.

**Tabel 5.12 Controller menambahkan data pegawai**

```
public function add_Karyawan ()
{
$this->model->addpegawai ();
include 'View/View_FormPegawai.php';
}
```

Controller pada Tabel 5.12 digunakan untuk menghubungkan antara *database* dengan halaman tampilan. Fungsi `add_Karyawan ()` digunakan

untuk menambahkan data pegawai yang telah dimasukan melalui *form* yang telah diisi pada `View_FormPegawai.php`. Kemudian dimasukan ke dalam *database* melalui fungsi `addpegawai ()`;

#### 5.2.4 Menampilkan Daftar Pegawai

**Tabel 5.13 Model menampilkan data pegawai**

```
public function getdaftarpegawai ()
{
    include "condatabase.php";
    $data_karyawan=array ();
    $query = mysqli_query ($koneksi, "SELECT * FROM data_karyawan
order by loker asc") or die (mysqli_error ());
    if (mysqli_num_rows ($query) == 0){
        echo "<b>Tidak ada data yang tersedia</b>";
    }else{
        while ($r = $query->fetch_assoc ()){
            $data_karyawan[] = $r;
        }return $data_karyawan;
    }
}
```

Untuk menampilkan daftar pegawai yang ada, program ini mengambil data pada tabel `data_karyawan` yang ada pada *database*, kemudian menyimpannya dalam array `$data_karyawan` setelah itu data akan ditampilkan dihalaman tampilan, seperti pada Tabel 5.13.

**Tabel 5.14 Contoller menampilkan data pegawai**

```
public function daftar_Karyawan ()
{
    $data_karyawan=$this->model->getdaftarpegawai ();
    $ARRAY = $this->koneksi ();
    include 'View/View_DaftarPegawai.php';
}
```

Fungsi `daftar_Karyawan ()` pada Tabel 5.14 digunakan untuk menghubungkan antara *database* dengan halaman tampilan, sehingga data – data yang telah diambil pada *database* tadi bisa ditampilkan.

**Tabel 5.15 View menampilkan data pegawai**

```
<table border="1" cellspacing="0" cellpadding="4">
<tr style="text-align:center;background-color:#ff8080">
<td> No </td>
<td>Nama</td>
<td>NIK</td>
```



```

        <td>Loker</td>
        <td>MAC</td>
        <td></td>
    </tr>
    <?php
        $no=1;
        foreach ($data_karyawan as $r){ ?>
        <tr style="text-align:center">
        <td><?php echo "$no" ?> </td>
        <td><?php echo $r['nama']?></td>
        <td><?php echo $r['nik'] ?></td>
        <td><?php echo $r['loker'] ?></td>
        <td><?php echo $r['mac2'] ?></td>
        <td>
            <form action="index.php" method="post"><input type="hidden"
name="edit" value="ok">
                <input type="hidden" name="nama" value="<?php echo
$r['nama'];?>">
                <input type="hidden" name="nik" value="<?php echo
$r['nik'];?>">
                <input type="hidden" name="loker" value="<?php echo
$r['loker'];?>">
                <input type="hidden" name="mac2" value="<?php echo
$r['mac2'];?>">
                <button class="waves-effect"
type="submit">edit</button></form>
            <form action="index.php" method="post"><input type="hidden"
name="hapus" value="ok"><input type="hidden" name="mac2"
value="<?php echo $r['mac2'];?>"><button class="waves-effect"
type="submit">delete</button></form>
        </td>
        </tr>
    <?php
        $no++;
    }
    ?>
</table>
</form>

```

Dapat dilihat pada Tabel 5.15, data yang tersimpan dalam `$data_karyawan` diinisialisasi menjadi `$r` kemudian data-data ditampilkan dalam bentuk tabel pada halaman `View_DaftarPegawai.php`.

### 5.2.5 Menghapus Data Pegawai

**Tabel 5.16 View delete data pegawai**

```
<form action="index.php" method="post"><input type="hidden"
name="hapus" value="ok"><input type="hidden" name="mac2"
value="<?php echo $r['mac2'];?>"><button class="waves-effect"
type="submit">delete</button></form>
```

Dapat dilihat pada Tabel 5.16, ketika menekan button hapus maka form akan mengirim post berisi alamat MAC yang akan dihapus pada *database* yaitu \$r['mac2'], alamat MAC ini lah yang menjadi paratemter data pegawai yang ingin dihapus, karena mac2 adalah *primery key* pada tabel data tersebut.

**Tabel 5.17 Model delete data pegawai**

```
public function deletepegawai ($mac2)
{
    include "condatabase.php";
    $deletepegawai = "DELETE FROM data_karyawan WHERE mac2 = '$mac2'";
    $queryhapus = mysqli_query ($koneksi, $deletepegawai) or die
    (mysqli_error ());
    if ($queryhapus) {
        header ('');
    } else{
        echo "Upss Something wrong..";
    }
}
```

Dapat dilihat pada Tabel 5.17, setelah mendapatkan alamat MAC data yang ingin dihapus maka program menjalankan fungsi query yang ada pada variable \$deletepegawai untuk menghapus data pegawai sesuai dengan alamat MAC yang dipilih.

**Tabel 5.18 Controller delete data pegawai**

```
public function delete_karyawan (){
    $this->model->deletepegawai ($_POST['mac2']);
    $this->daftar_Karyawan ();
}
```

Dapat dilihat pada Tabel 5.18, fungsi ini digunakan untuk menghubungkan antara *database* dengan halaman tampilan. Fungsi delete\_karyawan () akan menjalankan perintah deletepegawai (\$\_POST['mac2']) pada model, untuk menghapus data pegawai yang diinginkan pada *database*.

### 5.2.6 Edit Data Pegawai

**Tabel 5.19 Model menampilkan data pegawai yang akan diedit**

```
public function getpegawai ($mac2)
```

```

{
    include "condatabase.php";
    $getpegawai = "SELECT * FROM data_karyawan where mac2='$mac2'";
    $queryget = mysqli_query ($koneksi, $getpegawai) or die
    (mysqli_error ());
    if ($queryget) {
        header ('');
    } else{
        echo "Upss Something wrong..";
    }
}

```

Dapat dilihat pada Tabel 5.19, untuk mengubah data pegawai, maka yang harus dilakukan adalah mengambil data yang ingin diubah, parameter pengambilan data adalah \$mac2 dikarenakan \$mac2 adalah primery key dari tabel data\_karyawan yang ada pada *database*. Fungsi `getpegawai ($mac2)` digunakan untuk mengambil data pegawai yang ingin diubah pada *database*.

**Tabel 5.20 View menampilkan form edit**

```

<form action="index.php" method="POST" >
    <input type="hidden" name="update" value="ok">
    <table>
    <tr>
        <td>Nama</td>
        <td><input type="text" name="nama" value= "<?php echo
$_POST['nama']?>"></td>
    </tr>
    <tr>
        <td>NIK </td>
        <td><input type="text" name="nik" value= "<?php echo
$_POST['nik']?>"></td>
    </tr>
    <tr>
        <td>Loker</td>
        <td> <input type="text" name="loker" value= "<?php echo
$_POST['loker']?>"></td>
    </tr>
    <tr>
        <td>Mac</td>
        <td><input type="text" name="mac2" value= "<?php echo
$_POST['mac2']?>"></td>
    </tr>
    <tr>

```

```

<td></td>
    <td><input type="submit" name="Simpan" value="Simpan" >

</td>
</tr>
</table>
</form>

```

Dapat dilihat pada Tabel 5.20, data yang diambil pada *database* akan ditampilkan dalam *form*, sehingga *value* yang di tampilkan pada *form* adalah data yang berasal dari data pegawai yang kita pilih untuk di ubah. Setelah data dirubah, dan disimpan maka *form* akan mengirim *method post* yang berisi data yang telah diubah.

**Tabel 5.21 Model edit data pegawai**

```

public function updatepegawai ($mac2)
{
include "condatabase.php";
if (isset ($_POST['Simpan'])) {
    echo isset ($_POST['Simpan']);
    $nama      = $_POST['nama'];
    $nik       = $_POST['nik'];
    $loker     = $_POST['loker'];
    $mac2      = $_POST['mac2'];

    $querytambah = mysqli_query ($koneksi, "UPDATE data_karyawan
set nama='$nama', nik='$nik', loker='$loker' WHERE mac2='$mac2'")
or die (mysqli_error ());

    if ($querytambah) {
        header ('');
    } else {
        echo "Upss Something wrong..";
    }
}
}

```

Dapat dilihat pada Tabel 5.21, fungsi dari `updatepegawai ($mac2)` adalah untuk melakukan *update* data pegawai yang telah dirubah pada *database*. Setelah data yang diedit disimpan, maka akan melakukan pengecekan data pada `$_POST['Simpan']` jika terdapat data maka data tersebut akan disimpan dalam variable `$nama, $nik, $loker, $mac2` sesuai dengan data yang di *post* pada *form*.

**Tabel 5.22 Contoller edit data pegawai**

```

public function update_karyawan () {
    $this->model->updatepegawai ($_POST['mac2']);
    include 'index.php';
}

public function get_karyawan () {
    include 'View/View_UpdatePegawai.php';
}
}
}

```

Dapat dilihat pada Tabel 5.22, fungsi `update_karyawan ()` digunakan untuk menjalankan fungsi `updatepegawai ($_POST['mac2'])` pada *class* model. Fungsi `get_karyawan ()` digunakan untuk menampilkan data karyawan yang ingin diubah.

### 5.2.7 Tampil *History* Pegawai

**Tabel 5.23 Model menampilkan *history* kehadiran**

```

public function gethistorykehadiran ()
{
    include "condatabase.php";
    $data_history= array ();
    if (isset ($_POST['tambah'])) {
        $date= $_POST['tanggal'];
        var_dump ($date);

        $query = mysqli_query ($koneksi, "SELECT
data_karyawan.nama,data_karyawan.nik,data_karyawan.loker,pegawai
.mac,pegawai.date from pegawai, data_karyawan where pegawai.mac=
data_karyawan.mac2 and date='$date'" ) or die (mysqli_error ());

        if (mysqli_num_rows ($query) == 0) {
            echo "<b>Tidak ada data yang tersedia</b>";
        } else {

            while ($r = $query->fetch_assoc ()) {
                $data_history[] = $r;
            }
        }
        return $data_history;
    } else {
        $query = mysqli_query ($koneksi, "SELECT
data_karyawan.nama,data_karyawan.nik,data_karyawan.loker,pegawai
.mac,pegawai.date from pegawai, data_karyawan where pegawai.mac=

```

```

data_karyawan.mac2 order by pegawai.date desc ") or die
(mysql_error ());
if (mysqli_num_rows ($query) == 0){
    echo "<b>Tidak ada data yang tersedia</b>";
}
else{

    while ($r = $query->fetch_assoc ()){
        $data_history[] = $r;
    }
    return $data_history;
}
}

```

Dapat dilihat pada Tabel 5.23, pada saat menampilkan *history* pegawai program akan menampilkan seluruh data pegawai yang pernah hadir, jika ingin menampilkan pada tanggal tertentu saja maka program akan menjalankan `order by pegawai.date`, yang artinya menampilkan data pegawai yang hadir berdasarkan waktu yang telah dipilih.

**Tabel 5.24 View menampilkan *history* kehadiran**

```

<!DOCTYPE html>
<html>
<head>
    <title>ABSENSI SKJ</title>
</head>
<body>
<br>
    <div style='text-align:right;'>
        <form action="index.php" method="post"><input type="hidden"
name="history_ByNama" value="ok">
        <button class="waves-effect" type="submit">Cari Berdasarkan
Nama >>></button></form></div>
<center><h1 class="mt-5">HISTORY PRESENSI</h1><br></center>
<center>
<form action="index.php" method = "POST">
    <input type="hidden" name="getdate" value="getdate">
    <input type="date" name="tanggal">
    <input type="submit" name="tambah">
</form>
<table border="1" cellspacing="0" cellpadding="4">
    <tr style="text-align:center;background-color:#ff8080">
        <td> No </td>

```



```

        <td>Nama</td>
        <td>NIK</td>
        <td>Loker</td>
        <td>Tanggal</td>
    </tr>
<?php
$no=1;
foreach($data_history as $r){ ?>
    <tr style="text-align:center">
        <td><?php echo "$no" ?> </td>
        <td><?php echo $r['nama']?></td>
        <td><?php echo $r['nik'] ?></td>
        <td><?php echo $r['loker'] ?></td>
        <td><?php echo $r['date'] ?></td>
    </tr>
    <?php
    $no++;
    }?>
</tr></table>
</body>
</html>

```

Dapat dilihat pada Tabel 5.24, untuk menampilkan data pegawai yang hadir sesuai tanggal yang dipilih maka kita dapat mengisinya pada form yang ada pada View\_history.php

**Tabel 5.25 Contoller menampilkan *history* pegawai**

```

public function tampil_history ()
{
    $data_history=$this->model->gethistorykehadiran ();
    include 'View/View_History.php';
}

```

Dapat dilihat pada Tabel 5.25, data *form* yang dipilih pada halaman *view* akan dikirim ke model melalui *controller*, kemudian model akan mengambil data pada *database* dan ditampilkan kembali di halaman *view* melalui *controller*.

**Tabel 5.26 View *history* pegawai sesuai nama**

```

<!DOCTYPE html>
<html>
<head>
    <title>ABSENSI SKJ</title>

```

```

</head>
<body>
<br>
<center><h1 class="mt-5">HISTORY PRESENSI</h1><br>
<form action="index.php" method = "POST">
    <input type="hidden" name="caridata" value="caridata">
    <input type="text" name="carinama">
    <input type="submit" name="cari">
</form>
<table border="1" cellspacing="0" cellpadding="4">
<tr style="text-align:center;background-color:#ff8080">
<td> No </td>
    <td>Nama</td>
    <td>NIK</td>
    <td>Loker</td>
    <td>Tanggal dan Jam</td>
</tr>
<?php
$no=1;
foreach($hasilpencarian as $r){ ?>
<tr style="text-align:center">
<td><?php echo "$no" ?> </td>
    <td><?php echo $r['nama']?></td>
    <td><?php echo $r['nik'] ?></td>
    <td><?php echo $r['loker'] ?></td>
    <td><?php echo $r['date'] ?></td>
</tr>
<?php
$no++;
} ?>
</tr></table>
</body>
</html>

```

Pada saat memilih untuk menampilkan *history* pegawai berdasarkan nama, maka akan menampilkan halaman baru yaitu View\_History\_ByNama.php, halaman ini berisi daftar pegawai yang pernah hadir, dan sebuah *form* pencarian untuk mengisi nama pegawai yang ingin ditampilkan. Setelah mengisi *form* pencarian maka *form* akan mengirimkan *method* POST kehalaman index.php

seperti pada Tabel 5.26. Data yang dikirim diteruskan kehalaman *controller* pada fungsi `tampil_history_byNama()`, seperti pada Tabel 5.27.

**Tabel 5.27 Controller *history* pegawai sesuai nama**

```
public function tampil_history_byNama()
{
    $hasilpencarian=$this->model->gethistorybynama();
    include 'View/View_History_ByNama.php';
}
```

Pada halaman *controller* data yang dikirimkan kemudian diteruskan pada halaman *model*, seperti fungsi `$hasilpencarian=$this->model->gethistorybynama()`, yang ada pada Tabel 5.27.

**Tabel 5.28 Model *history* pegawai sesuai nama**

```
public function gethistorybynama()
{
    include "condatabase.php";
    $hasilpencarian= array();
    if(isset($_POST['cari'])){
        $carinama= $_POST['carinama'];
        $query = mysqli_query($koneksi, "SELECT
data_karyawan.nama,data_karyawan.nik,data_karyawan.loker,pegawai
.mac,pegawai.date from pegawai, data_karyawan where pegawai.mac=
data_karyawan.mac2 and data_karyawan.nama like '%$carinama%'" ) or die
(mysqli_error());
        if(mysqli_num_rows($query) == 0){
            echo "<b>Tidak ada data yang tersedia</b>";
        }else{
            while($r = $query->fetch_assoc()){
                $hasilpencarian[] = $r;
            }
            return $hasilpencarian;
        }else{
            $query = mysqli_query($koneksi, "SELECT
data_karyawan.nama,data_karyawan.nik,data_karyawan.loker,pegawai
.mac,pegawai.date from pegawai, data_karyawan where pegawai.mac=
data_karyawan.mac2 order by pegawai.date desc ") or die
(mysqli_error());
            if(mysqli_num_rows($query) == 0){
                echo "<b>Tidak ada data yang tersedia</b>";
            }else{
```

```
while($r = $query->fetch_assoc()){  
    $hasilpencarian[] = $r;  
}  
}return $hasilpencarian;  
}  
  
}
```

Data yang diteruskan oleh conteoller kemudian diproses pada fungsi `gethistorybynama()`. Pada fungsi tersebut diambil isi dari method POST yang dikirimkan melalui *form* pada halaman *View*, kemudian disimpan pada *variable* baru `$carinama`, *variable* ini yang menjadi patokan dalam menampilkan *history* sesuai dengan nama pegawai yang ingin dicari. Terlihat pada *variable* `$query` pada Tabel 5.28 yang berisi fungsi untuk mengambil data pada *database*. *Sourcecode* `data_karyawan.nama like '%$carinama%'` memiliki arti, data yang harus diambil pada *database* harus sesuai dengan huruf yang tersimpan pada *variable* `$carinama`. Jika tidak ada yang sama maka akan tetap menampilkan seluruh data pegawai yang pernah hadir.



## BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dan analisis berguna untuk mengetahui apakah kinerja dari *sistem* pendeteksi kehadiran yang telah dirancang sudah sesuai dengan kebutuhan dan perancangan yang telah dijelaskan pada bab - bab sebelumnya. Pada penelitian ini Proses pengujian dibagi menjadi dua yaitu pengujian fungsional dan pengujian performa.

### 6.1 Pengujian Fungsional

Pengujian fungsional merupakan pengujian secara *black-box* yang digunakan untuk mengetahui apakah sistem sudah berjalan dengan baik dan benar. Pengujian dilakukan untuk mengetahui interaksi antara sistem pendeteksi kehadiran, router, dan *database*. Pengujian dianggap berhasil jika semua fungsi pada sistem yang telah dibangun tidak menunjukkan adanya kesalahan saat dilakukan pengujian. Skenario pengujian yang dilakukan dapat dilihat pada Tabel 6.1.

**Tabel 6.1 Skenario pengujian fungsional**

Kode	Fungsi	Skenario
PF_001	Sistem melakukan koneksi ke router.	<ol style="list-style-type: none"> <li>1. Sistem melakukan koneksi ke router menggunakan API untuk RouterOS melalui ip 192.168.88.1.</li> <li>2. Sistem terkoneksi ke router.</li> </ol>
PF_002	Sistem mendeteksi kehadiran pegawai	<ol style="list-style-type: none"> <li>1. Sistem melakukan pencocokan data yang masuk ke <i>database</i> dari router dengan data alamat MAC yang sudah ada pada <i>database</i>.</li> <li>2. Sistem menampilkan data pegawai yang hadir.</li> </ol>
PF_003	Sistem mendeteksi lokasi pegawai yang hadir.	<ol style="list-style-type: none"> <li>1. Sistem terkoneksi pada router.</li> <li>2. Sistem membaca <i>interface</i> dimana <i>smartphone</i> terkoneksi.</li> <li>3. Sistem mencocokkan data <i>interface</i> tempat <i>smartphone</i> pegawai terkoneksi dengan data pegawai yang ada pada <i>database</i>.</li> <li>4. Sistem berhasil menampilkan lokasi pegawai.</li> </ol>
PF_004	Sistem menambah data pegawai ke <i>database</i> .	<ol style="list-style-type: none"> <li>1. Sistem menampilkan form untuk memasukan data pegawai.</li> </ol>

		<p>2. Sistem menyimpan data pegawai yang diinputkan ke dalam form kemudian memasukkannya pada <i>database</i>.</p> <p>3. Sistem menyimpan data pegawai ke dalam <i>database</i>.</p>
PF_005	Sistem menghapus data pegawai pada <i>database</i> .	<p>1. Sistem akan menampilkan daftar data pegawai yang ada pada <i>database</i>.</p> <p>2. Memilih opsi delete pada daftar data pegawai.</p> <p>3. Data pegawai yang dipilih berhasil dihapus.</p>
PF_006	Sistem melakukan edit data pegawai.	<p>1. Sistem akan menampilkan daftar data pegawai yang ada pada <i>database</i>.</p> <p>2. Memilih edit pada daftar data pegawai.</p> <p>3. Sistem menampilkan form yang berisi data pegawai yang ingin diedit.</p> <p>4. Sistem menyimpan data pegawai yang dirubah melalui form, kemudian melakukan pembaruan pada <i>database</i> sesuai data yang telah dirubah pada form.</p>
PF_007	Sistem menampilkan daftar data pegawai yang ada pada <i>database</i> .	<p>1. Memilih daftar pegawai pada menu.</p> <p>2. Sistem menampilkan daftar data pegawai yang ada.</p>
PF_008	Sistem menampilkan <i>history</i> kehadiran pegawai sesuai tanggal atau nama yang ingin ditampilkan	<p>1. Memilih <i>history</i> pada menu.</p> <p>2. Sistem menampilkan semua daftar pegawai yang hadir.</p> <p>3. Memilih tanggal kehadiran atau memilih Menampilkan Berdasarkan Nama.</p> <p>4. Jika memilih Menampilkan Berdasarkan Nama, maka akan menampilkan layar baru berisi <i>form</i> pencarian dan daftar pegawai yang pernah hadir.</p> <p>5. Melakukan pengisian nama pada form.</p> <p>6. Menampilkan berdasarkan nama.</p> <p>7. Jika tidak maka memilih tanggal pada <i>form</i> tanggal.</p>

		8. Menampilkan daftar pegawai yang hadir sesuai dengan tanggal yang dipilih.
PF_009	Sistem melakukan insert data berupa alamat MAC ke dalam <i>database</i> .	1. Melakukan insert ke dalam <i>database</i> berupa alamat MAC. 2. Data alamat MAC dimasukan ke dalam <i>database</i> .

Tabel 6.2 Hasil pengujian fungsional

Kode	Hasil	Keterangan
PF_001	Sistem berhasil terkoneksi dengan router.	Berhasil
PF_002	Sistem berhasil mendeteksi kehadiran pegawai	Berhasil
PF_003	Sistem berhasil mendeteksi lokasi pegawai yang hadir.	Berhasil
PF_004	Sistem berhasil menambahkan data pegawai yang hadir.	Berhasil
PF_005	Sistem berhasil menghapus data pegawai pada <i>database</i> .	Berhasil
PF_006	Sistem berhasil melakukan edit data pegawai.	Berhasil
PF_007	Sistem berhasil menampilkan daftar data pegawai yang ada pada <i>database</i> .	Berhasil
PF_008	Sistem berhasil menampilkan <i>history</i> kehadiran pegawai sesuai tanggal atau nama yang ingin ditampilkan.	Berhasil
PF_009	Sistem berhasil melakukan insert data berupa alamat MAC ke dalam <i>database</i> .	Berhasil



## 6.2 Pengujian Performa

Pengujian performa dilakukan untuk menguji kinerja dari sistem. Pengujian berfokus pada waktu yang diperlukan sistem mendeteksi kehadiran pegawai, dimana dibagi menjadi 4 yaitu waktu yang diperlukan sistem terkoneksi pada router dan membaca alamat MAC dan *interface smartphone* yang ada pada router, waktu yang diperlukan sistem untuk menampilkan data pegawai yang hadir atau yang diperlukan untuk mendeteksi perpindahan lokasi *smartphone* antara router atau *access point*, dan tingkat skalabilitas jumlah perangkat yang ditangani oleh sistem tiap detik, dan waktu yang diperlukan sistem untuk mendeteksi kehadiran dari pertama terkoneksi ke router sampai dibaca oleh sistem. Pengujian akan dilakukan menggunakan Wireshark.

**Tabel 6.3 Skenario pengujian performa**

Kode	Parameter	Keterangan
PP_001	Waktu yang diperlukan sistem membaca alamat MAC dan lokasi <i>smartphone</i> yang ada pada router .	Kondisi sistem terkoneksi pada router mikrotik, kemudian melakukan 18 kali percobaan, yang pertama dengan 1 perangkat yang terkoneksi, kemudian 5 perangkat, dan 10 perangkat, percobaan dilakukan pada router mikrotik, <i>access point</i> 1 dan <i>access point</i> 2. Semua hasil percobaan akan di <i>capture</i> oleh wireshark dan akan dilakukan analisis.
PP_002	Waktu yang diperlukan sistem untuk mendeteksi perpindahan lokasi <i>smartphone</i> .	Menganalisis menggunakan wireshark dengan kondisi sistem terkoneksi pada router mikrotik, dan terdapat <i>smartphone</i> yang terkoneksi pada router atau <i>access point</i> . Kemudian akan dilakukan percobaan perangkat berpindah dari satu <i>access point</i> ke <i>access point</i> lainnya, dari router ke <i>access point</i> dan juga sebaliknya. Kemudian akan dihitung rata-rata waktu dari dua kali percobaan untuk menghitung waktu untuk mendeteksi perpindahan lokasi <i>smartphone</i> .
PP_003	Waktu yang diperlukan sistem untuk mendeteksi kehadiran dari pertama terkoneksi ke router sampai	Menganalisis menggunakan wireshark dengan kondisi pertama sistem langsung terkoneksi pada router mikrotik, kedua terkoneksi pada <i>access point</i> 1, dan yang ke tiga terkoneksi pada <i>access point</i> 2.

	dibaca oleh sistem .	
PP_004	Jumlah <i>smartphone</i> yang dapat dibaca oleh sistem dari router < 1 detik .	Menganalisis menggunakan wireshark, dengan variasi <i>smartphone</i> yang terkoneksi yaitu 1 (satu) <i>smartphone</i> , 5 (lima) <i>smartphone</i> secara bersamaan, dan 10 (sepuluh) <i>smartphone</i> secara bersamaan, percobaan dilakukan dengan perangkat terkoneksi pada router mikrotik.

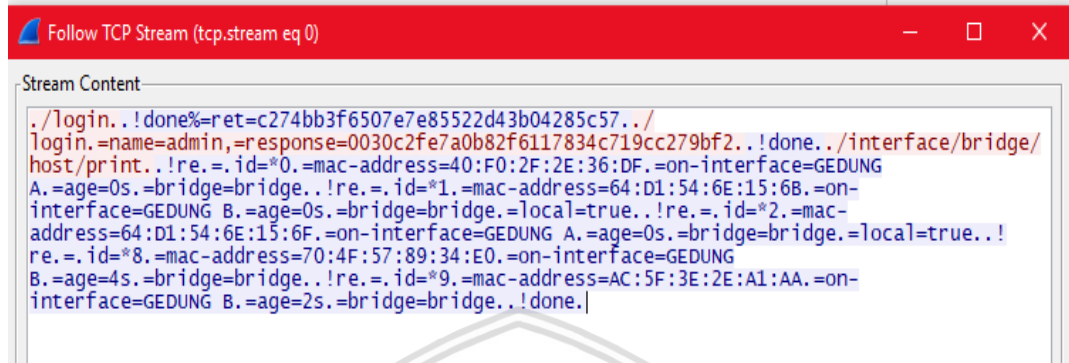
Pengujian ini diawali dengan memastikan router mikrotik dalam keadaan menyala dan tersambung dengan kedua *access point*, kemudian menjalankan XAMPP agar sistem dapat bekerja, dan memastikan bahwa sistem terkoneksi pada router mikrotik. Setelah semua sudah dilakukan, selanjutnya melakukan pengujian performa pertama yaitu menghitung waktu yang diperlukan sistem untuk membaca alamat MAC dan lokasi *smartphone* yang ada pada router (PP\_001) . Pengujian ini melakukan 18 kali percobaan, yang pertama dengan 1 perangkat yang terkoneksi, kemudian 5 perangkat, dan 10 perangkat, percobaan dilakukan pada router mikrotik, *access point* 1 dan *access point* 2 masing-masing percobaan dilakukan sebanyak 2 kali percobaan untuk mendapatkan hasil yang akurat.

Setiap percobaan akan *capture* oleh wireshark, dari hasil capture wireshark tersebut akan dihitung selisih waktu dari awal sistem melakukan koneksi pada router sampai sistem membaca alamat MAC dan lokasi pegawai. Caranya adalah dengan melihat selisih waktu yang diambil pada hasil capture wireshark yaitu *time since reference of first frame* yang sudah ditampilkan juga pada tabel time yang dapat dilihat pada Gambar 6.1.

No.	Time	Source	Destination	Protocol	Length	Time delta from previous displayed frame	Destination Port
3	3.17382000	192.168.88.254	192.168.88.1	TCP	66	1.676399000	8728
4	3.25654500	192.168.88.1	192.168.88.254	TCP	66	0.082725000	51780
5	3.25671400	192.168.88.254	192.168.88.1	TCP	54	0.000169000	8728
6	3.25687300	192.168.88.254	192.168.88.1	TCP	61	0.000159000	8728
7	3.25786800	192.168.88.1	192.168.88.254	TCP	54	0.000995000	51780
8	3.25790600	192.168.88.254	192.168.88.1	TCP	55	0.000038000	8728
9	3.25920100	192.168.88.1	192.168.88.254	TCP	54	0.001295000	51780
10	3.31063700	192.168.88.1	192.168.88.254	TCP	99	0.051436000	51780
11	3.31099600	192.168.88.254	192.168.88.1	TCP	61	0.000359000	8728
12	3.31205300	192.168.88.1	192.168.88.254	TCP	54	0.001057000	51780
13	3.31211000	192.168.88.254	192.168.88.1	TCP	112	0.000057000	8728
14	3.31312600	192.168.88.1	192.168.88.254	TCP	54	0.001016000	51780
15	3.31793900	192.168.88.1	192.168.88.254	TCP	61	0.004813000	51780
16	3.31814700	192.168.88.254	192.168.88.1	TCP	83	0.000208000	8728
17	3.35145900	192.168.88.1	192.168.88.254	TCP	54	0.033312000	51780
18	3.35151900	192.168.88.254	192.168.88.1	TCP	55	0.000060000	8728
19	3.35249800	192.168.88.1	192.168.88.254	TCP	54	0.000979000	51780
20	3.37049900	192.168.88.1	192.168.88.254	TCP	535	0.018001000	51780
21	3.41094800	192.168.88.254	192.168.88.1	TCP	54	0.040449000	8728
23	3.63578400	192.168.88.254	192.168.88.1	TCP	54	0.140808000	8728
26	3.68994000	192.168.88.1	192.168.88.254	TCP	54	0.000001000	51780
27	3.69010500	192.168.88.254	192.168.88.1	TCP	54	0.000165000	8728

**Gambar 6.1 Hasil capture wireshark pengujian PP\_001**

Komunikasi yang terjadi pada nomer 3 sampai nomer 27 pada Gambar 6.1 tersebut adalah komunikasi antara sistem dan router mikrotik, untuk melihat komunikasi apa saja yang terjadi dapat dilihat pada Gambar 6.2 .

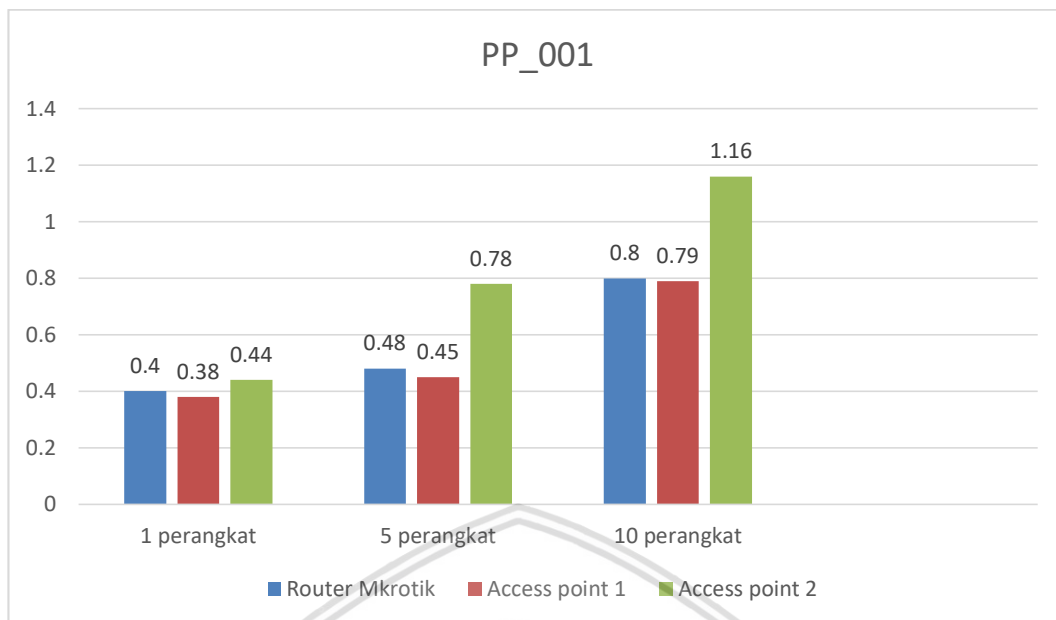


**Gambar 6.2 Follow TCP stream pengujian PP\_001**

Selisih waktu pada nomer 3 dan nomer 27 pada Gambar 6.1 menjadi hasil dari waktu yang dibutuhkan sistem untuk membaca alamat MAC dan lokasi yang ada pada router, alasannya karena dari nomer 3 sampai nomer 27 merupakan proses mulai dari sistem melakukan koneksi dengan router sampai router memberikan data ke sistem. Hasil dari pengujian P\_001 dapat dilihat pada Tabel 6.4.

**Tabel 6.4 Hasil pengujian PP\_001**

Waktu yang dibutuhkan oleh sistem (s)			
	Router Mikrotik	Access point 1	Access point 2
1 Perangkat	0,3941	0,4619	0,3699
	0.4063	0,2983	0.5150
Rata-rata	0,4002	0.3801	0,4424
5 Perangkat	0,5372	0,5925	0,8398
	0,4107	0,3088	0,7186
Rata-rata	0.4739	0,4506	0,7792
10 Perangkat	0,8013	0,8405	1,1340
	0,8015	0,7506	1,1903
Rata-rata	0,8014	0,7955	1,1621



**Gambar 6.3 Grafik hasil pengujian**

Dari hasil pengujian pada Tabel 6.4 bahwa jumlah perangkat yang terkoneksi berpengaruh terhadap waktu yang diperlukan sistem untuk membaca alamat MAC dan lokasi perangkat pada router, perbedaan tersebut dapat dilihat pada Gambar 6.3. Posisi perangkat terkoneksi juga dapat mempengaruhi lama waktu sistem dalam membaca alamat MAC.

Pengujian yang kedua adalah menguji waktu yang diperlukan untuk sistem mendeteksi perpindahan lokasi *smartphone*. Skenario pengujian hampir sama dengan pengujian PP\_001, hanya saja perangkat yang sudah terkoneksi akan melakukan perpindahan lokasi, dalam artian adalah berpindah koneksi pada router atau *access point* lainnya. Pengujian dilakukan sebanyak 6 kali, 2 kali percobaan pada perpindahan perangkat dari Gedung A (router mikrotik) menuju Gedung B (*access point* 1), 2 kali percobaan dari Gedung B ke Gedung C (*access point* 3), dan Gedung C ke Gedung A. Untuk menghitung waktu yang diperlukan sistem untuk mendeteksi perpindahan lokasi *smartphone* didapatkan melalui selisih dari waktu saat perangkat masih terkoneksi pada lokasi sebelumnya hingga ke lokasi sekarang. Selisih waktu tersebut dapat dilihat pada Gambar 6.4.

No.	Time	Source	Destination	Protocol
69	7.80629000	192.168.88.254	192.168.88.1	TCP
70	7.80748900	192.168.88.1	192.168.88.254	TCP
71	7.80760600	192.168.88.254	192.168.88.1	TCP
72	7.80795800	192.168.88.254	192.168.88.1	TCP
73	7.80895100	192.168.88.1	192.168.88.254	TCP
74	7.80898200	192.168.88.254	192.168.88.1	TCP
75	7.81004100	192.168.88.1	192.168.88.254	TCP
76	7.81256900	192.168.88.1	192.168.88.254	TCP
77	7.81277100	192.168.88.254	192.168.88.1	TCP
78	7.84687200	192.168.88.1	192.168.88.254	TCP
79	7.84694900	192.168.88.254	192.168.88.1	TCP
80	7.84795300	192.168.88.1	192.168.88.254	TCP
81	7.85166100	192.168.88.1	192.168.88.254	TCP
82	7.85188300	192.168.88.254	192.168.88.1	TCP
83	7.88683200	192.168.88.1	192.168.88.254	TCP
84	7.88689100	192.168.88.254	192.168.88.1	TCP
85	7.88784100	192.168.88.1	192.168.88.254	TCP
86	7.89148100	192.168.88.1	192.168.88.254	TCP
87	7.93289600	192.168.88.254	192.168.88.1	TCP
88	8.09023200	192.168.88.254	192.168.88.1	TCP
89	8.09138100	192.168.88.1	192.168.88.254	TCP
90	8.09145900	192.168.88.254	192.168.88.1	TCP

Gambar 6.4 Hasil capture wireshark pengujian PP\_002

Pada pengujian ini *smartphone* yang digunakan untuk melakukan uji coba memiliki alamat MAC yaitu AC:5F:3E:2E:A1:AA. Pada Gambar 6.4 terlihat bahwa *smartphone* yang terkoneksi berlokasi pada Gedung A.

No.	Time	Source	Destination	Protocol
108	13.2030420	192.168.88.254	192.168.88.1	TCP
109	13.2519720	192.168.88.1	192.168.88.254	TCP
110	13.2521490	192.168.88.254	192.168.88.1	TCP
111	13.2523040	192.168.88.254	192.168.88.1	TCP
112	13.2533900	192.168.88.1	192.168.88.254	TCP
113	13.2534390	192.168.88.254	192.168.88.1	TCP
114	13.2545350	192.168.88.1	192.168.88.254	TCP
115	13.2679220	192.168.88.1	192.168.88.254	TCP
116	13.2689010	192.168.88.254	192.168.88.1	TCP
117	13.3069270	192.168.88.1	192.168.88.254	TCP
118	13.3070040	192.168.88.254	192.168.88.1	TCP
119	13.3080300	192.168.88.1	192.168.88.254	TCP
120	13.3117990	192.168.88.1	192.168.88.254	TCP
121	13.3120770	192.168.88.254	192.168.88.1	TCP
122	13.3469410	192.168.88.1	192.168.88.254	TCP
123	13.3470270	192.168.88.254	192.168.88.1	TCP
124	13.3481670	192.168.88.1	192.168.88.254	TCP
125	13.3525060	192.168.88.1	192.168.88.254	TCP
126	13.3940990	192.168.88.254	192.168.88.1	TCP
127	13.5362090	192.168.88.1	192.168.88.254	TCP
128	13.5374350	192.168.88.1	192.168.88.254	TCP
129	13.5375050	192.168.88.254	192.168.88.1	TCP

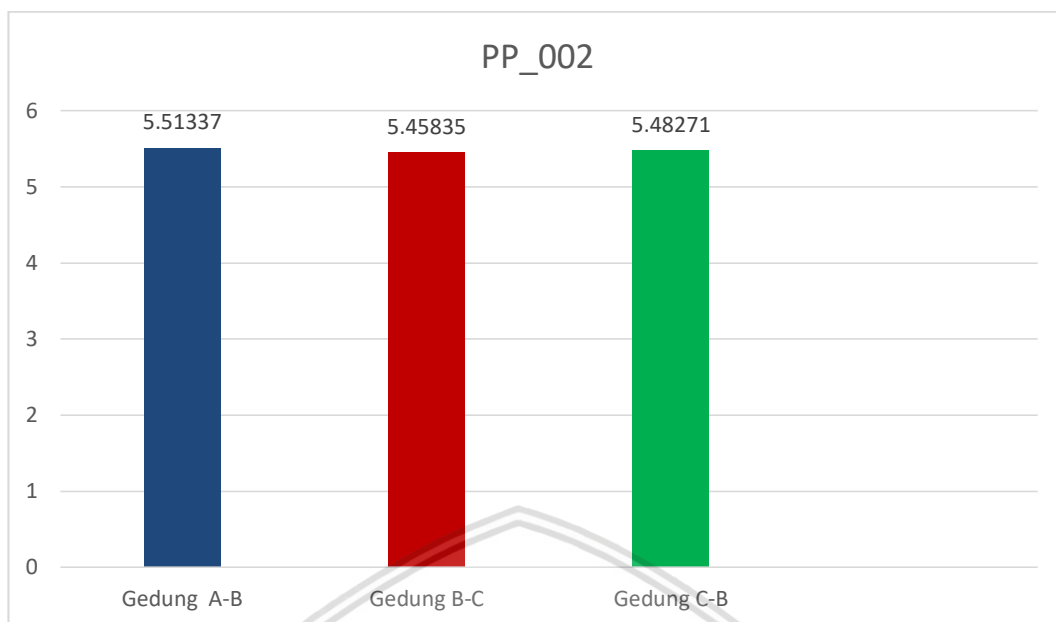
Gambar 6.5 Hasil capture wireshark pengujian PP\_002

Pada Gambar 6.5 bisa kita lihat bahwa *smartphone* yang terkoneksi telah berpindah lokasi pada Gedung B. Untuk mengetahui berapa waktu yang diperlukan agar sistem dapat mendeteksi perpindahan lokasi *smartphone* adalah dengan menghitung selisih waktu saat *smartphone* masih berada pada lokasi sebelumnya dengan lokasi sekarang. Hasil dari pengujian P\_002 dapat dilihat pada Tabel 6.5.

Tabel 6.5 Hasil pengujian PP\_002

No.	Gedung A – Gedung B	Gedung B- Gedung C	Gedung C – Gedung A
1	5,44605 s	5,42556 s	5,51117 s
2	5,58069 s	5,49115 s	5,45426 s
Rata-rata	5,51337 s	5,45835 s	5,48271 s





**Gambar 6.6 Grafik hasil pengujian PP\_002**

Pada Tabel 6.5 Hasil pengujian PP\_002, dapat dilihat bahwa rata-rata sistem untuk mendeteksi perpindaahn lokasi sebuah perangkat yaitu kurang lebih 5 detik. Gambar 6.6 menunjukan bahwa rata-rata waktu yang dibutuhkan sistem untuk mendeteksi perpindahan lokasi antar router atau *access point* hampir sama.

Pengujian yang ketiga adalah PP\_003 yaitu menghitung waktu yang diperlukan sistem untuk mendeteksi kehadiran dari pertama terkoneksi ke router sampai dibaca oleh sistem dari router . Skenario pengujian ini sama dengan pengujian PP\_001 hanya saja dikarenakan yang dihitung adalah waktu sejak pertama *smartphone* terkoneksi sampai di dibaca oleh sistem, maka waktu saat pertama *smartphone* terkoneksi pada router sampai sistem melakukan *login* ke router dan menampilkan data pada web juga dihitung. Untuk mengetahui waktu saat pertama terkoneksi pada router sampai sistem melakukan *login* pada router adalah seperti pada Gambar 6.7 .

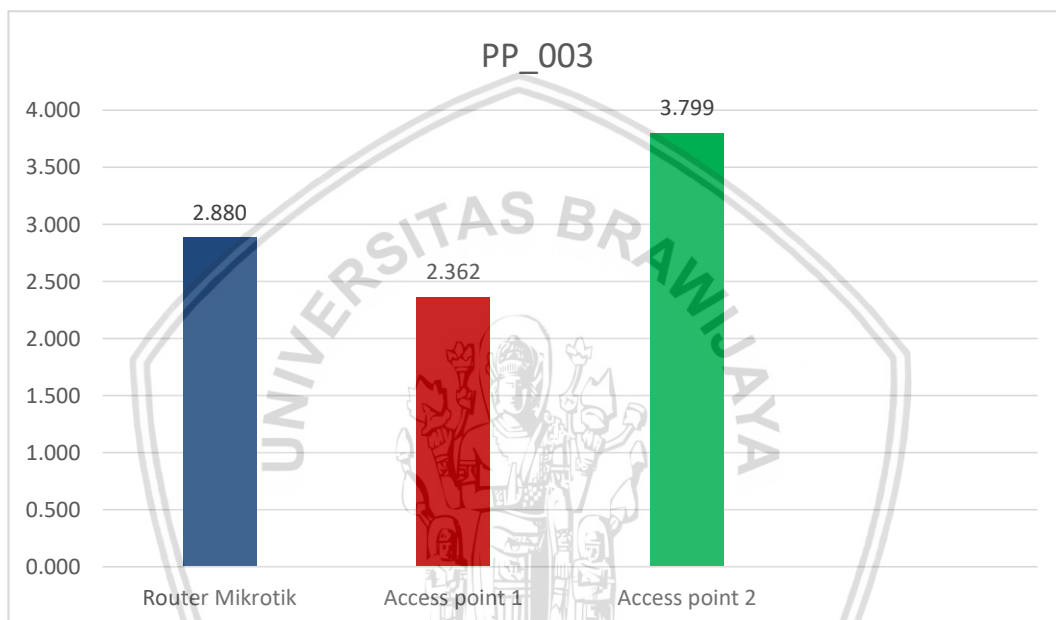
Jan/28/2018 01:29:15	memory	wireless, info	AC:5F:3E:2E:A1:AA@GEDUNG A: connected
Jan/28/2018 01:29:16	memory	dhcp, info	defconf deassigned 192.168.88.246 from AC:5F:3E:2E:A1:AA
Jan/28/2018 01:29:16	memory	dhcp, info	defconf assigned 192.168.88.246 to AC:5F:3E:2E:A1:AA
Jan/28/2018 01:29:17	memory	system, info, account	user admin logged in from 192.168.88.247 via api

**Gambar 6.7 Hasil capture wireshark pengujian PP\_002**

Pada Gambar 6.7 terlihat bahwa dibutuhkan waktu 2 detik dari awal *smartphone* terkoneksi sampai sistem melakukan login terhadap router, kemudian waktu tersebut ditambah lagi dengan waktu saat sistem melakukan *login* dan membaca data pada router seperti pada Pengujian PP\_001. Kemudian data tersebut juga ditambahkan dengan waktu yang dibutuhkan sistem untuk melakukan *insert* data pada *database*, dan mengambil data dalam *database* (Lampiran 3). Hasil dari pengujian PP\_003 dapat dilihat pada Tabel 6.6 .

**Tabel 6.6 Hasil pengujian PP\_003**

Waktu yang dibutuhkan sistem (s)			
No.	Router Mikrotik	Access point 1	Access point 2
1	2,860	1,792	3,837
2	4,917	2,933	3,761
Rata-rata	2,888	2,362	3,799



**Gambar 6.8 Grafik hasil pengujian PP\_003**

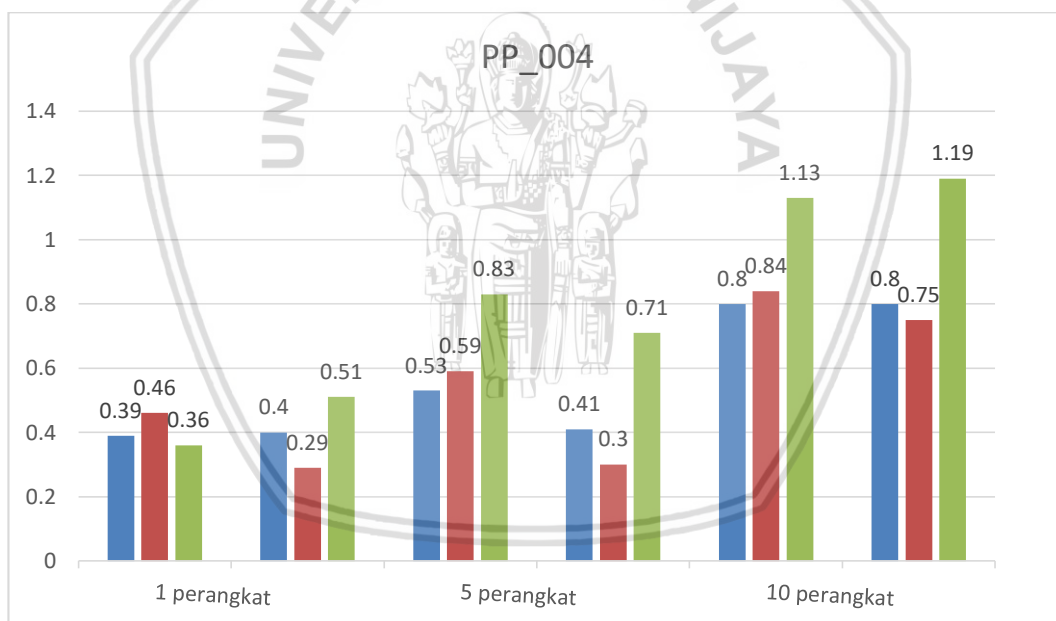
Terlihat pada Gambar 6.8 bahwa rata-rata yang dibutuhkan sistem berbeda-beda, ini dikarenakan pada sistem diatur untuk melakukan pemeriksaan pada router setiap 5 detik sekali, sehingga ketika melakukan percobaan ada yang cepat terdeteksi dan ada yang lambat, tergantung pengecekan dari sistem.

Pengujian yang terakhir adalah menghitung jumlah *smartphone* yang dapat dibaca oleh sistem dari router < 1 detik, yaitu pengujian PP\_004. Pengujian ini sama dengan pengujian PP\_001, sehingga kita menggunakan hasil dari pengujian 1 untuk dianalisis pada pengujian PP\_004 ini. Berikut hasil pengujian dijabarkan pada Tabel 6.7.



Tabel 6.7 Hasil pengujian PP\_004

Waktu yang dibutuhkan sistem (s)						
	Router Mikrotik		Access point 1		Access point 2	
	Rata-rata waktu (s) <1	Rata-rata waktu (s) >1	Rata-rata waktu (s) <1	Rata-rata waktu (s) >1	Rata-rata waktu (s) <1	Rata-rata waktu (s) >1
1 Perangkat	0,3941	-	0,4619	-	0,3699	-
	0,4063	-	0,2983	-	0,5150	-
5 Perangkat	0,5372	-	0,5925	-	0,8398	-
	0,4107	-	0,3088	-	0,7186	-
10 Perangkat	0,8013	-	0,8405	-	-	1,1340
	0,8015	-	0,7506	-	-	1,1903



Gambar 6.9 Grafik hasil pengujian PP\_004

Dari hasil pengujian pada Tabel 6.7, terlihat pengujian 1 buah *smartphone* dapat dibaca oleh sistem kurang dari 1 detik, begitu juga pada pengujian 5 buah *smartphone*. Pada pengujian 10 *smartphone* bisa dilihat pada Gambar 6.9, saat terkoneksi langsung pada mikrotik, sistem mampu membaca kurang dari 1 detik begitu juga pada saat terkoneksi pada *access point* 1, tetapi pada saat terkoneksi pada *access point* 2, sistem mendeteksi lebih dari 1 detik. Hal mungkin saja terjadi, dan banyak faktor yang mempengaruhi hal tersebut, seperti spesifikasi router atau *access point*, kondisi perangkat keras (komputer atau laptop) saat menjalankan sistem.

## BAB 7 PENUTUP

### 7.1 Kesimpulan

Pada bagian ini menguraikan mengenai kesimpulan yang didapat berdasarkan hasil dari seluruh proses penelitian yang telah dilakukan.

Berdasarkan hasil tersebut, maka didapatkan kesimpulan sebagai berikut:

1. API untuk RouterOS merupakan sebuah fitur yang tersedia pada router mikrotik digunakan agar dapat terintegrasi dengan sistem eksternal. Hal ini memungkinkan pengguna dapat mengembangkan perangkat lunak sendiri yang dapat berkomunikasi dengan RouterOS untuk mendapatkan informasi, konfigurasi dan pengelolaan perangkat tersebut. Dengan adanya API pengguna dapat mengembangkan berbagai macam perangkat lunak sesuai dengan kebutuhan pengguna itu sendiri. API dapat diimplementasikan ke berbagai macam bahasa pemrograman salah satunya adalah PHP.
2. Sistem pendeteksi kehadiran pegawai ini dirancang menggunakan bahasa pemrograman PHP . Sistem mengimplementasikan PHP API untuk berkomunikasi dan mengambil data yang ada pada router mikrotik. Data berupa alamat MAC yang telah diambil oleh sistem dari router mikrotik disimpan dalam array kemudian di masukan ke dalam *database*. Sistem ini juga menggunakan XAMPP sebagai server dan penyimpanan data. Pada penelitian kali ini sistem dirancang memiliki 2 buah *access point* dan 1 buah router mikrotik, dengan kedua *access point* di bridge pada router mikrotik, sehingga data-data pada perangkat yang masuk pada *access point* juga dapat terbaca oleh router mikrotik.
3. Proses pengolahan data dimulai dari sistem mendeteksi alamat MAC perangkat yang terkoneksi pada router mikrotik, data tersebut disimpan dalam array, kemudian dimasukan ke dalam *database*. Data berupa alamat MAC kemudian dihubungkan dengan data karyawan yang berada pada *database* yang sama, jika alamat MAC sesuai dengan data karyawan maka sistem akan menampilkan nama pegawai pada daftar hadir. Untuk mendeteksi lokasi pegawai, sistem membaca *interface smartphone* yang terkoneksi pada router mikrotik, kemudian dihubungkan dengan data pegawai yang hadir, jika sama maka sistem akan menampilkan lokasi pegawai yang hadir tersebut.
4. Dari hasil pengujian fungsional, diketahui bahwa sistem pendeteksi kehadiran pegawai yang dibangun, telah berfungsi seperti apa yang telah dijelaskan dan diuraikan pada bab sebelumnya yaitu rekayasa kebutuhan, serta perancangan dan implementasi.

Kemudian dari hasil pengujian performa didapatkan kesimpulan sebagai berikut :

- a. Lama waktu yang dibutuhkan sistem untuk membaca alamat MAC pada *smartphone* yang terkoneksi pada router atau *access point* berbanding lurus dengan jumlah *smartphone* yang terkoneksi. Sehingga semakin sedikit *smartphone* yang terkoneksi dan dibaca oleh sistem maka akan semakin sedikit waktu yang dibutuhkan, begitupula sebaliknya.
- b. Router atau *access point* di mana *smartphone* terkoneksi juga berpengaruh terhadap lama waktu proses yang dilakukan oleh sistem, ini terlihat ketika 10 user terkoneksi pada router mikrotik dan *access point* 1, sistem mampu membaca alamat MAC dari router kurang dari 1 detik, sedangkan pada *access point* 2 lebih dari 1 detik.
- c. Lama waktu yang dibutuhkan oleh sistem untuk mendeteksi kehadiran pegawai dari awal *smartphone* terkoneksi pada router sangat sangat berpengaruh pada kapan sistem melakukan pemeriksaan pada router mikrotik, pemeriksaan tersebut dilakukan setiap 5 detik.
- d. Lama waktu sistem untuk membaca perpindahan lokasi tempat *smartphone* terkoneksi relatif sama pada setiap perpindahannya.

## 7.2 Saran

Berdasarkan kesimpulan penelitian yang telah diuraikan sebelumnya, maka ada beberapa saran untuk penelitian berikutnya yang terkait dengan topik pada penelitian ini dijabarkan sebagai berikut:

- a. Sistem ini hanya dapat membaca data dari sebuah router mikrotik, penelitian berikutnya dapat dikembangkan agar dapat membaca dari router lainnya, mengingat harga router mikrotik yang cukup mahal.
- b. Sistem ini belum memiliki keamanan yang bagus, penelitian berikutnya dapat menambah tingkat keamanan sehingga menghindari adanya manipulasi data seperti pemalsuan alamat MAC dan lain-lain.
- c. Pengujian performa dapat dilakukan dengan menguji aspek lain, dari sudut pandang yang lebih luas seperti *reliability* dan *security*.

## DAFTAR PUSTAKA

- Adhikara, F. (2014). Analisis dan Perancangan Sistem Presensi Berbasis Global Positioning System (GPS) Pada Android 4.x. Indonesia: Universitas Esa Unggul.
- Al-Husainy, M., A., F. (2014). Mac Address as a Key for Data Encryption, *International Journal of Computer Science and Information Security*.
- Boots Group Plc., (2003). *Corporate social responsibility*. [online] Boots Group Plc. Diakses 15 September, 2017 dari <http://www.boots-plc.com/information/info.asp?id1=1a>
- Dermawan, D., Permana, D. (2013). *Desain dan Pemrograman Website*. Bandung: PT. Remaja Rosdakarya.
- Dganalakshmi, N., Komar, S.G., (2017). Aadhaar Based Biometric Attendance System Using Wireless *Fingerprint* Terminals. *IEEE 7th International Advance Computing Conference*. 651-652.
- Fathansyah, 2013. *Basis Data*. Bandung: Informatika.
- Grag, U., Verma, p., Moudgil, Y., S., Sharma, S. (2012). *MAC and Logical Addressing (A review Studay)*. *International Journal of Engineering Research and Applications (IJERA)*, 476-477.
- Imam, K.S., Rizal, R.I., Kridalukman, R. (2016). *Optimizing MySQL Database System on Information Systems Research, Publications and Community Service*. IEEE.
- Lan, Y., Sun, Y., Liu, S., & Ma, Z. (2017). A real-time network traffic analysis and QoS management platform. China: IEEE.
- Materializecss. (2017). Templates. Diakses 17 Oktober, 2017 dari <http://materializecss.com/getting-started.html>
- Mikrotik. (2017). API PHP Class. Diakses 3 September, 2017 dari [https://wiki.mikrotik.com/wiki/API\\_PHP\\_class](https://wiki.mikrotik.com/wiki/API_PHP_class)
- Patil, C.S., Karhe, R.R., Jain, M.R. (2014). *Student Attendance Recording System Using Face Recognition with GSM Based*. *International Journal of Research in Advent Technology*. 373-374.
- Putu, I.A.E.P., (2013). *Handbook Jaringan Komputer*. Bandung: Informatika.
- Stoitsov, G., Rongelov, V. (2014). One implementation of API interface for RouterOS. *TEM Journal*, Volume 3. 148-149.
- Stoitsov, G., Rongelov, V. (2014). One implementation of API interface for RouterOS. *TEM Journal*, Volume 3. 150-151.
- W3schools. (2017). SQL Tutorial. Diakses 7 September, 2017 dari <https://www.w3schools.com/sql/default.asp>